

Guide de l'utilisateur

AWS CodeBuild



Version de l'API 2016-10-06

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeBuild: Guide de l'utilisateur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service qui n'appartient pas à Amazon, de toute manière susceptible de créer une confusion chez les clients ou de toute manière dénigrant ou discréditant Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce que c'est AWS CodeBuild ?	1
Comment courir CodeBuild	1
Tarification pour CodeBuild	3
Comment puis-je commencer CodeBuild ?	3
Concepts	3
Comment CodeBuild fonctionne	3
Étapes suivantes	5
Prise en main	6
Mise en route à l'aide de la console	6
Étape 1 : Création du code source	7
Étape 2 : Création du fichier buildspec	10
Étape 3 : Création de deux compartiments S3	12
Étape 4 : Charger le code source et le fichier buildspec	13
Étape 5 : Créer le projet de génération	15
Étape 6 : Exécuter la génération	17
Étape 7 : Afficher les informations de génération résumées	17
Étape 8 : Afficher les informations de génération détaillées	18
Étape 9 : Obtenir l'artefact de sortie de génération	19
Étape 10 : Supprimer les compartiments S3	20
Récapitulatif	21
Prise en main à l'aide de l' AWS CLI	21
Étape 1 : Création du code source	22
Étape 2 : Création du fichier buildspec	25
Étape 3 : Création de deux compartiments S3	28
Étape 4 : Charger le code source et le fichier buildspec	28
Étape 5 : Créer le projet de génération	30
Étape 6 : Exécuter la génération	34
Étape 7 : Afficher les informations de génération résumées	36
Étape 8 : Afficher les informations de génération détaillées	39
Étape 9 : Obtenir l'artefact de sortie de génération	41
Étape 10 : Supprimer les compartiments S3	42
Récapitulatif	43
Exemples basés sur des cas d'utilisation	44
Échantillons multiservices	45

Exemple Amazon ECR	46
Exemple Amazon EFS	53
AWS CodePipeline échantillons	59
AWS Config échantillon	70
Exemple de notifications de génération	72
Exemple de badges de génération	88
Créez un projet de construction avec des badges de construction	89
Badges Access AWS CodeBuild Build	92
Publier CodeBuild des badges de construction	93
CodeBuild statuts des badges	93
Exemple de rapport de test	93
Exécutez l'exemple de rapport de test	94
Exemples Docker pour CodeBuild	101
Exemple Docker dans une image personnalisée	101
Exemple de serveur de génération d'images Docker	104
Exemple de build Windows Docker	107
Exemple « Publier une image Docker sur Amazon ECR »	110
Registre privé avec AWS Secrets Manager échantillon	119
Hébergement d'une sortie de génération dans un compartiment S3	123
Échantillon d'entrées et de sorties multiples	126
Créez un projet de construction avec plusieurs entrées et sorties	127
Création d'un projet sans source	130
Versions d'environnement d'exécution dans l'exemple de fichier buildspec	131
Mettre à jour la version d'exécution dans le fichier buildspec	131
Spécification de deux environnements d'exécution	136
Exemple de version source	140
Spécifiez une version de GitHub référentiel avec un ID de validation	142
Spécifiez une version GitHub du référentiel avec une référence et un ID de validation	143
Exemples de référentiels sources tiers	144
Exécutez l'exemple Bitbucket	145
Exécutez l'exemple de serveur d' GitHub entreprise	151
Exécutez l' GitHub exemple de pull request et de filtre Webhook	158
Tutoriel : signature de code Apple avec Fastlane lors de CodeBuild l'utilisation de S3 pour le stockage de certificats	162
Tutoriel : Signature de code Apple avec Fastlane dans CodeBuild Utilisation GitHub pour le stockage de certificats	169

Définissez les noms des artefacts au moment de la construction	175
Exécuter des exemples Windows	178
Exécutez les exemples Windows	178
Structure de répertoire	179
F# et .NET Framework	180
Visual Basic et .NET Framework	180
Fichiers	180
F# et .NET Framework	180
Visual Basic et .NET Framework	185
Planification d'une génération	199
Référence des spécifications de génération	202
Nom de fichier buildspec et emplacement de stockage	202
Syntaxe d'un fichier buildspec	203
version	206
run-as	207
env	207
proxy	212
phases	213
rapports	217
artefacts	219
cache	225
Exemple de fichier buildspec	227
Versions de fichier buildspec	231
Référence des spécifications de construction par lots	231
lot	232
batch/build-graph	232
batch/build-list	235
batch/build-matrix	238
batch/build-fanout	240
Référence sur les environnements de génération	242
Images Docker fournies par CodeBuild	243
Obtenir la liste des images Docker actuelles	243
EC2 calculer des images	244
Images de calcul Lambda	245
Images déconseillées CodeBuild	250
Runtimes disponibles	251

Versions d'environnement d'exécution	268
Modes et types de calcul de l'environnement de création	273
À propos du calcul	273
À propos des types d'environnement de capacité réservée	274
À propos des types d'environnements à la demande	327
Shells et commandes dans les environnements de génération	339
Variables d'environnement dans les environnements de génération	341
Tâches en arrière-plan dans les environnements de génération	347
Construisez des projets	348
Création d'un projet de génération	348
Prérequis	349
Création d'un projet de génération (console)	349
Création d'un projet de génération (AWS CLI)	372
Création d'un projet de génération (AWS SDKs)	393
Création d'un projet de génération (CloudFormation)	394
Création d'une règle de notification	394
Modifier les paramètres du projet de construction	397
Modification des paramètres d'un projet de génération (console)	397
Modification des paramètres d'un projet de génération (AWS CLI)	423
Modification des paramètres d'un projet de génération (AWS SDKs)	424
Jetons d'accès multiples	424
Étape 1 : Création d'un secret ou d'une CodeConnections connexion dans le Gestionnaire de Secrets	425
Étape 2 : Accorder au rôle IAM CodeBuild du projet l'accès aux secrets de Secrets Manager	426
Étape 3 : Configuration du Gestionnaire de Secrets Manager ou CodeConnections des jetons	428
Options de configuration supplémentaires	432
Supprimer des projets de construction	435
Suppression d'un projet de génération (console)	436
Suppression d'un projet de génération (AWS CLI)	436
Suppression d'un projet de génération (AWS SDKs)	437
Obtenir un projet de construction public URLs	437
Partagez des projets de construction	438
Partagez un projet	439
Services connexes	442

Accédez à des projets partagés	442
Annuler le partage d'un projet partagé	443
Identifier un projet partagé	443
Autorisations de projet partagés	443
Tag : projets de construction	444
Ajouter une balise à un projet	445
Afficher les balises d'un projet	447
Modifier les balises d'un projet	448
Supprimer une balise d'un projet	449
Utilisez des coureurs	450
GitHub Actions	451
GitLab coureurs	474
Coureur Buildite	489
Utiliser des webhooks	512
Bonnes pratiques d'utilisation des webhooks	513
Événements du webhook Bitbucket	514
GitHub webhooks mondiaux et organisationnels	528
GitHub webhooks manuels	535
GitHub événements webhook	537
GitLab webhooks de groupe	554
GitLab webhooks manuels	560
GitLab événements webhook	561
Webhooks manuels Buildkite	576
Approbation des commentaires par pull request	578
Afficher les détails du projet de construction	583
Affichage des détails d'un projet de génération (console)	584
Affichage des détails d'un projet de génération (AWS CLI)	584
Affichage des détails d'un projet de génération (AWS SDKs)	586
Afficher les noms des projets de construction	587
Affichage d'une liste de noms de projet de génération (console)	587
Affichage d'une liste de noms de projet de génération (AWS CLI)	587
Affichage d'une liste de noms de projet de génération (AWS SDKs)	589
Builds	590
Exécuter les builds manuellement	591
Exécuter un build localement	592
Exécution d'une génération (console)	595

Exécution d'une génération (AWS CLI)	596
Exécuter une compilation par lots (AWS CLI)	603
Lancement automatique de l'exécution de générations (AWS CLI)	605
Arrêt automatique de l'exécution de générations (AWS CLI)	606
Exécution d'une génération (AWS SDKs)	607
Exécuter des builds sur le calcul Lambda	607
Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda	608
Et si l'image sélectionnée n'inclut pas les outils dont j'ai besoin ?	608
Dans quelles régions le AWS Lambda calcul CodeBuild est-il possible ?	609
Limites du AWS Lambda calcul	609
Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java	610
Créez une application React d'une seule page avec CodeBuild Lambda Node.js	614
Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python	617
Exécutez des builds sur des flottes à capacité réservée	621
Création d'un parc de capacités réservées	623
Bonnes pratiques	625
Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ?	625
Comment fonctionne le calcul basé sur les attributs ?	626
Puis-je spécifier manuellement une instance Amazon EC2 pour mon parc ?	626
Quelles régions prennent en charge les flottes à capacité réservée ?	627
Comment configurer une flotte de macOS à capacité réservée ?	627
Comment configurer une Amazon Machine Image (AMI) personnalisée pour un parc de capacités réservées ?	628
Limitations des flottes à capacité réservée	630
Propriétés du parc à capacité réservée	630
Échantillons de capacité réservée	635
Exécuter des builds par lots	637
Rôle de sécurité	637
Types de build par lots	638
Mode de rapport par lots	642
En savoir plus	642
Exécuter des tests parallèles	642
Support dans AWS CodeBuild	644
Activer l'exécution de tests en parallèle dans les versions par lots	646
Utilisez la codebuild-tests-run commande CLI	647

Utilisez la codebuild-glob-search commande CLI	650
À propos du fractionnement des tests	652
Fusionnez automatiquement les rapports de construction individuels	653
Exemples d'exécution de tests parallèles	655
Constitutions de cache	666
Mise en cache d'Amazon S3	667
Mise en cache locale	673
Spécifier un cache local	675
Déboguer les versions	677
Déboguer les builds avec Sandbox CodeBuild	678
Déboguer les builds avec le gestionnaire de session	678
Déboguer les builds avec Sandbox CodeBuild	678
Déboguer les builds avec le gestionnaire de session	708
Suppression de générations	713
Suppression de générations (AWS CLI)	713
Suppression de générations (AWS SDKs)	714
Réessayer les builds manuellement	715
Réessayer une compilation manuellement (console)	715
Réessayer une compilation manuellement (AWS CLI)	715
Réessayer une compilation manuellement (AWS SDKs)	716
Réessayez de compiler automatiquement	716
Réessayer une compilation automatiquement (console)	717
Réessayer une compilation automatiquement (AWS CLI)	717
Réessayer automatiquement un build (AWS SDKs)	718
Arrêter les builds	718
Arrêt d'une génération (console)	718
Arrêt d'une génération (AWS CLI)	719
Arrêt d'une génération (AWS SDKs)	719
Arrêter les compilations par lots	720
Arrêter une génération par lots (console)	720
Arrêter une construction par lots (AWS CLI)	721
Arrêter une construction par lots (AWS SDKs)	721
Le déclencheur se construit automatiquement	721
Création de déclencheurs de génération	722
Modification des déclencheurs de génération	725
Affichage des détails d'une génération	729

Affichage des détails d'une génération (console)	729
Affichage des détails d'une génération (AWS CLI)	730
Affichage des détails d'une génération (AWS SDKs)	730
Transitions des phases de génération	730
Afficher la version IDs	731
Afficher la liste des versions IDs (console)	731
Afficher la liste des build IDs (AWS CLI)	732
Afficher la liste des build IDs (AWS CLI) par lots	733
Afficher la liste des build IDs (AWS SDKs)	734
Afficher la version IDs pour un projet de construction	735
Afficher la liste des builds IDs pour un projet de build (console)	735
Afficher la liste des versions IDs pour un projet de construction (AWS CLI)	735
Afficher la liste des compilations par lots IDs pour un projet de construction (AWS CLI)	737
Afficher la liste des versions IDs pour un projet de construction (AWS SDKs)	738
Rapports d'essais	739
Création de rapports de test	740
Création de rapports sur la couverture du code	742
Création d'un rapport sur la couverture du code	742
Découvrez les rapports automatiquement	744
Configurer la découverte automatique des rapports à l'aide de la console	745
Configurer la découverte automatique des rapports à l'aide des variables d'environnement du projet	745
Groupes de rapports	746
Créer un groupe de rapports	747
Attribution des noms des groupes de rapports	753
Partager des groupes de rapports	754
Spécification des fichiers de test	760
Spécification des commandes de test	761
Baliser un groupe de rapports	761
Mise à jour d'un groupe de rapports	768
Cadres de test	771
Configurez Jasmine	771
Configurer Jest	774
Configurer pytest	775
Configurez RSpec	776
Affichage des rapports de test	777

Affichage des rapports de test d'une génération	777
Affichage des rapports de test d'un groupe de rapports	778
Afficher les rapports de test dans votre AWS compte	778
Autorisations relatives aux rapports de test	778
Rôle IAM pour les rapports de test	779
Autorisations pour les opérations de rapports de test	781
Exemples d'autorisations de rapport de test	781
Statuts des rapports de test	782
Prise en charge de VPC	784
Cas d'utilisation	784
Les meilleures pratiques pour VPCs	785
Limites de VPCs	786
Autorisez l'accès à Amazon VPC dans vos projets CodeBuild	786
Résoudre les problèmes de configuration de votre VPC	787
Utilisation de points de terminaison de VPC	788
Avant de créer des points de terminaison de VPC	788
Créez des points de terminaison VPC pour CodeBuild	789
Créez une politique de point de terminaison VPC pour CodeBuild	790
Utiliser un serveur proxy CodeBuild géré	790
Configurer une configuration de proxy géré pour les flottes de capacité réservée	791
Gérez un parc à capacité CodeBuild réservée	792
Utilisation d'un serveur proxy	793
Configuration des composants nécessaires à l'exécution CodeBuild sur un serveur proxy ...	794
Exécuter CodeBuild sur un serveur proxy explicite	797
Exécuter CodeBuild sur un serveur proxy transparent	801
Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy	803
CloudFormation Modèle VPC	805
Journalisation et surveillance	811
Enregistrer les appels CodeBuild d'API	811
À propos AWS CodeBuild des informations contenues dans CloudTrail	811
À propos des entrées du fichier AWS CodeBuild journal	813
Surveillez les versions	815
CloudWatch métriques	816
CloudWatch mesures d'utilisation des ressources	818
CloudWatch dimensions	820
CloudWatch alarmes	820

Afficher les CodeBuild métriques	821
Afficher les indicateurs d'utilisation des CodeBuild ressources	823
Créez des CodeBuild alarmes dans CloudWatch	827
Sécurité	829
Protection des données	830
Chiffrement des données	831
Gestion des clés	832
Confidentialité du trafic	833
Gestion des identités et des accès	833
Présentation de la gestion des accès	833
Utilisation de politiques basées sur l'identité	838
AWS CodeBuild référence aux autorisations	868
Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources	875
Affichage des ressources dans la console	879
Validation de conformité	880
Résilience	881
Sécurité de l'infrastructure	881
Accès au fournisseur de source	881
Créez et stockez un jeton dans un secret du Gestionnaire de Secrets	882
GitHub et accès GitHub aux serveurs d'entreprise	885
Accès à Bitbucket	897
GitLab accès	907
Prévention du cas de figure de l'adjoint désorienté entre services	913
Rubriques avancées	916
Permettre aux utilisateurs d'interagir avec CodeBuild	916
CodeBuild Autoriser l'interaction avec d'autres AWS services	924
Chiffrer les sorties de build	932
Interagissez à CodeBuild l'aide du AWS CLI	934
Référence des commandes en ligne	935
AWS SDKs et référence des outils	937
Support AWS SDKs et outils pour AWS CodeBuild	937
Travailler avec AWS SDKs	938
Spécifiez le CodeBuild point de terminaison	939
Spécifiez le AWS CodeBuild point de terminaison (AWS CLI)	940
Spécifiez le AWS CodeBuild point de terminaison (AWS SDK)	940
À utiliser CodeBuild avec CodePipeline	942

Conditions préalables	944
Création d'un pipeline (console)	945
Création d'un pipeline (AWS CLI)	950
Ajouter une action de construction	955
Ajouter une action de test	959
Utilisation CodeBuild avec Codecov	962
Intégration de Codecov dans un projet de génération	963
À utiliser CodeBuild avec Jenkins	966
Configurer Jenkins	966
Installer le plug-in	967
Utiliser le plugin	967
Utilisation CodeBuild avec des applications sans serveur	969
Ressources connexes	970
Avis aux tiers	970
1) image Docker de base — Windowsservercore	971
2) Image Docker basée sur Window—choco	972
3) Image Docker basée sur Windows : git --version 2.16.2	972
4) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools	973
5) Image Docker basée sur Windows : nuget.commandline --version 4.5.1	977
7) Image Docker basée sur Windows : netfx-4.6.2-devpack	977
8) Image Docker basée sur Windows — visualfsharptools, version 4.0	979
9) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies	979
10) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1	983
11) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx	988
12) Image Docker basée sur Windows : dotnet-sdk	989
Utiliser les clés de CodeBuild condition comme variables de rôle de service IAM	989
AWS CodeBuild clés de condition	990
Appliquez les paramètres de connectivité VPC à vos projets et à vos flottes	991
Empêchez les modifications non autorisées des spécifications de construction du projet	992
Limitez les types de calcul pour vos builds	992
Réglages variables de l'environnement de contrôle	993
Utiliser des variables dans les noms des clés de condition	994
Vérifier l'existence d'attributs dans les demandes d'API	995
Exemples de code	996
Principes de base	996

Actions	997
Résolution des problèmes	1014
Apache Maven génère des artefacts de référence à partir du mauvais répertoire	1015
Les commandes de génération s'exécutent en tant que racine par défaut	1017
Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains. Caractères anglais	1017
Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store	1018
Impossible d'accéder au filtre de branche dans la CodeBuild console	1019
Impossible d'afficher la réussite ou l'échec de la génération	1020
État du build non communiqué au fournisseur source	1020
Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019	1021
Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures	1021
Erreur : accès refusé lors de la tentative de téléchargement du cache	1022
Erreur : « BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE » lors de l'utilisation d'une image de génération personnalisée	1022
Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces	1023
Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération	1024
Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction	1025
Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl »	1026
Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération	1027
Erreur : « Git clone failed: Unable to access 'your-repository-URL': SSL certificate problem: Self signed certificate »	1027
Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération	1028
Erreur : « This build image requires selecting at least one runtime version. »	1028
Erreur : « QUEUED: INSUFFICIENT_SUBNET » lorsqu'une génération dans une file d'attente échoue	1029

Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie »	1030
Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied»	1030
Erreur : « Unable to locate credentials »	1031
RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy	1032
Le shell bourne (sh) doit exister dans les images de génération	1034
Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération	1034
Erreur : « Impossible de vérifier JobWorker l'identité »	1034
La construction n'a pas pu démarrer	1035
Accès aux GitHub métadonnées dans les versions mises en cache localement	1035
AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3...	1035
Erreur : « Vos informations d'identification ne disposent pas d'une ou de plusieurs étendues de privilèges requises » lors de la création d'un CodeBuild projet avec CodeConnections	1036
Erreur : « Désolé, aucun terminal n'a été demandé, impossible d'obtenir des informations » lors de la compilation avec la commande d'installation d'Ubuntu	1037
Quotas	1039
Quotas de service	1039
Autres limites	1046
Construisez des projets	1046
Construit	1046
Flottes informatiques	1047
Rapports	1048
Étiquettes	1049
Historique de la documentation	1050
Mises à jour antérieures	1078
.....	mx civ

Qu'est-ce que c'est AWS CodeBuild ?

AWS CodeBuild est un service de création entièrement géré dans le cloud. CodeBuild compile votre code source, exécute des tests unitaires et produit des artefacts prêts à être déployés. CodeBuild élimine le besoin de provisionner, de gérer et de dimensionner vos propres serveurs de construction. Ce service fournit des environnements de génération prépackagés pour les langages de programmation et outils de génération couramment utilisés, comme Apache Maven, Gradle, etc. Vous pouvez également personnaliser les environnements de construction CodeBuild pour utiliser vos propres outils de génération. CodeBuild évolue automatiquement pour répondre aux demandes de construction en période de pointe.

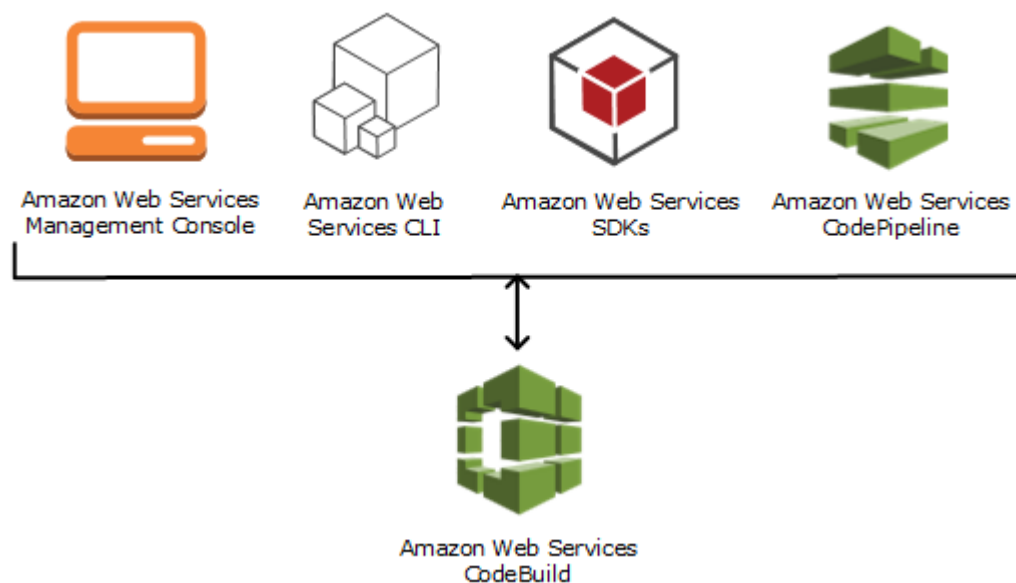
CodeBuild fournit les avantages suivants :

- Entièrement géré : vous CodeBuild n'avez plus besoin de configurer, de corriger, de mettre à jour et de gérer vos propres serveurs de build.
- À la demande : CodeBuild évolue à la demande pour répondre à vos besoins de construction. Vous ne payez que pour le nombre de minutes de génération que vous consommez.
- Prêt à l'emploi : CodeBuild fournit des environnements de construction préconfigurés pour les langages de programmation les plus courants. Il vous suffit de pointer sur votre script de génération pour démarrer votre première génération.

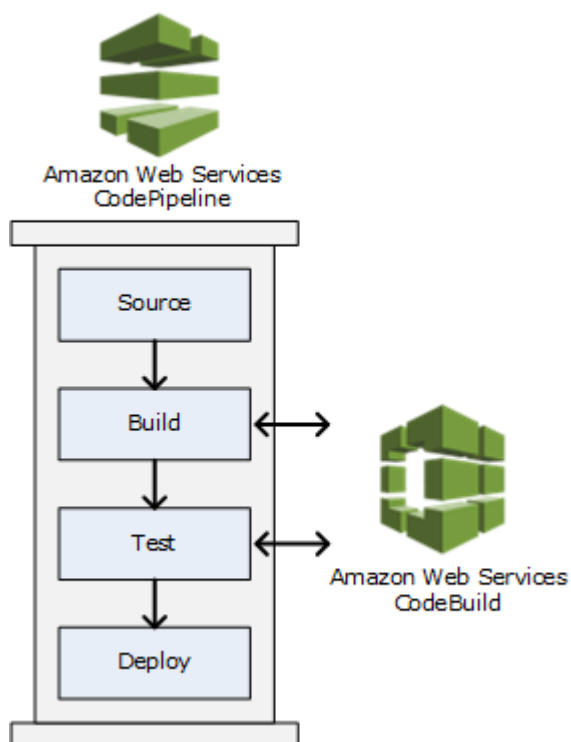
Pour de plus amples informations, veuillez consulter [AWS CodeBuild](#).

Comment courir CodeBuild

Vous pouvez utiliser la AWS CodePipeline console AWS CodeBuild or pour exécuter CodeBuild. Vous pouvez également automatiser le fonctionnement de CodeBuild en utilisant le AWS Command Line Interface (AWS CLI) ou le AWS SDKs.



Comme le montre le schéma suivant, vous pouvez ajouter CodeBuild une action de construction ou de test à la phase de construction ou de test d'un pipeline dans AWS CodePipeline. AWS CodePipeline est un service de livraison continue que vous pouvez utiliser pour modéliser, visualiser et automatiser les étapes nécessaires à la publication de votre code. Cela inclut la génération de votre code. Un pipeline est une structure de workflow qui décrit la progression des modifications de code d'un processus de publication.



À utiliser CodePipeline pour créer un pipeline puis ajouter une action de CodeBuild génération ou de test, voir [À utiliser CodeBuild avec CodePipeline](#). Pour plus d'informations CodePipeline, consultez le [guide de AWS CodePipeline l'utilisateur](#).

La CodeBuild console permet également de rechercher rapidement vos ressources, telles que des référentiels, des projets de création, des applications de déploiement et des pipelines. Choisissez Go to ressource (Accéder à la ressource) ou appuyez sur la touche /, puis saisissez le nom de la ressource. Toutes les correspondances s'affichent dans la liste. Les recherches ne sont pas sensibles à la casse. Vous pouvez uniquement consulter les ressources que vous êtes autorisé à afficher. Pour de plus amples informations, veuillez consulter [Affichage des ressources dans la console](#).

Tarification pour CodeBuild

Pour obtenir des informations, veuillez consulter [Tarification CodeBuild](#).

Comment puis-je commencer CodeBuild ?

Nous vous recommandons d'effectuer les étapes suivantes :

1. Pour CodeBuild en savoir plus, lisez les informations dans [Concepts](#).
2. Faites des essais CodeBuild dans un exemple de scénario en suivant les instructions de [Mise en route à l'aide de la console](#).
3. CodeBuild Utilisez-le dans vos propres scénarios en suivant les instructions de [Planification d'une génération](#).

AWS CodeBuild concepts

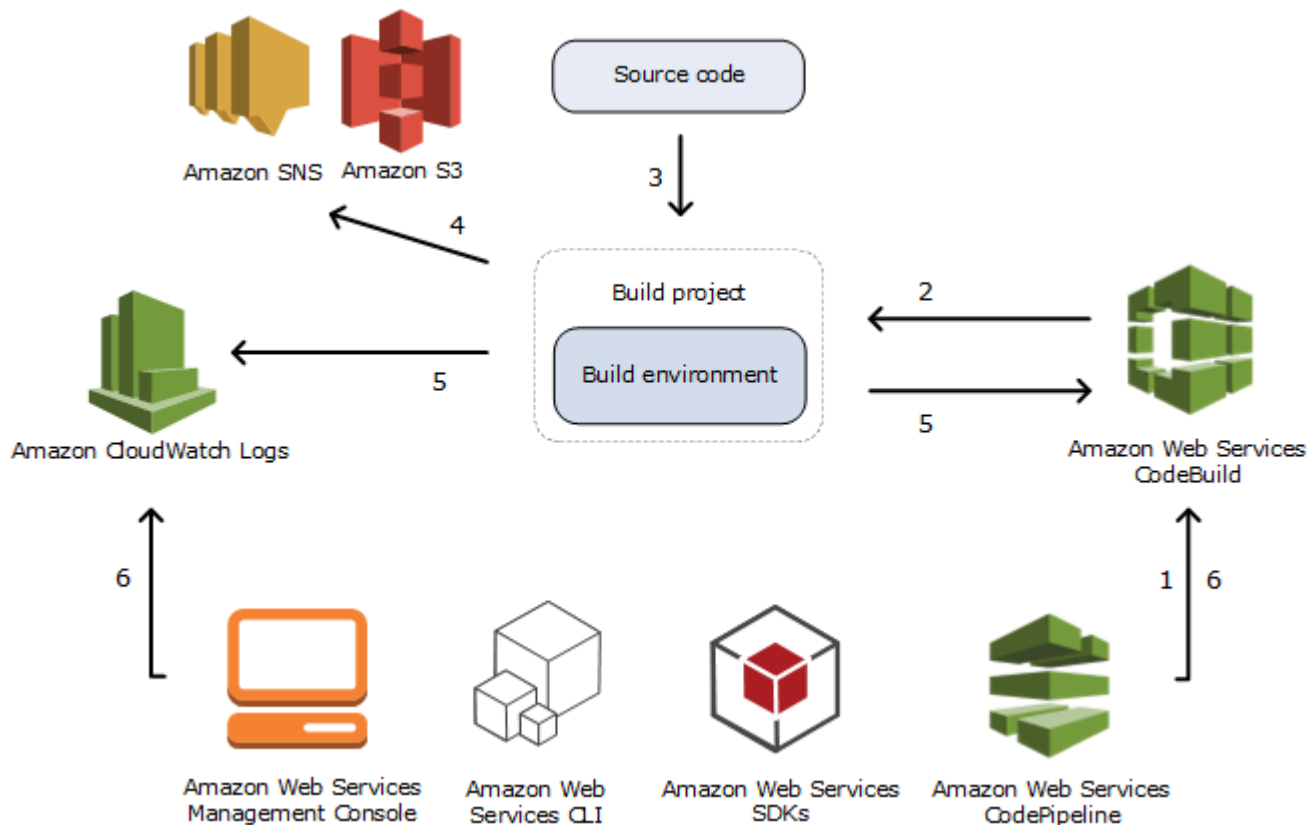
Les concepts suivants sont importants pour comprendre le CodeBuild fonctionnement.

Rubriques

- [Comment CodeBuild fonctionne](#)
- [Étapes suivantes](#)

Comment CodeBuild fonctionne

Le schéma suivant montre ce qui se passe lorsque vous exécutez une compilation avec CodeBuild :



1. En entrée, vous devez CodeBuild fournir un projet de construction. Un projet de build inclut des informations sur la façon d'exécuter une compilation, notamment sur l'emplacement du code source, sur l'environnement de compilation à utiliser, sur les commandes de compilation à exécuter et sur l'emplacement où stocker le résultat de la compilation. Un environnement de génération représente une combinaison de système d'exploitation, d'exécution d'un langage de programmation et d'outils CodeBuild utilisés pour exécuter une compilation. Pour en savoir plus, consultez :
 - [Création d'un projet de génération](#)
 - [Référence sur les environnements de génération](#)
2. CodeBuild utilise le projet de construction pour créer l'environnement de construction.
3. CodeBuild télécharge le code source dans l'environnement de construction, puis utilise la spécification de construction (buildspec), telle que définie dans le projet de construction ou incluse directement dans le code source. Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

4. S'il existe une sortie de génération, l'environnement de génération charge sa sortie dans un compartiment S3. L'environnement de génération peut également effectuer des tâches que vous spécifiez dans les spécifications de construction (par exemple, envoyer des notifications de compilation à une rubrique Amazon SNS). Pour obtenir un exemple, consultez [Exemple de notifications de génération](#).
5. Pendant que le build est en cours d'exécution, l'environnement de build envoie des informations à Amazon CloudWatch Logs CodeBuild et à Amazon.
6. Pendant l'exécution de la compilation, vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs obtenir des informations de compilation résumées CodeBuild et détaillées à partir d'Amazon CloudWatch Logs. AWS CLI Si vous avez l'habitude d'exécuter des builds, vous pouvez obtenir des informations de build limitées auprès de CodePipeline.

Étapes suivantes

Maintenant que vous en savez plus AWS CodeBuild, nous vous recommandons les étapes suivantes :

1. Faites des essais CodeBuild dans un exemple de scénario en suivant les instructions de [Mise en route à l'aide de la console](#).
2. CodeBuild Utilisez-le dans vos propres scénarios en suivant les instructions de [Planification d'une génération](#).

Commencer avec CodeBuild

Dans les didacticiels suivants, vous AWS CodeBuild allez créer une collection d'exemples de fichiers d'entrée de code source dans une version déployable du code source.

Les deux didacticiels ont les mêmes entrées et résultats, mais l'un utilise la AWS CodeBuild console et l'autre utilise le AWS CLI.

Important

Nous vous déconseillons d'utiliser votre compte AWS root pour suivre ce didacticiel.

Rubriques

- [Commencer à AWS CodeBuild utiliser la console](#)
- [Commencer à AWS CodeBuild utiliser le AWS CLI](#)

Commencer à AWS CodeBuild utiliser la console

Dans ce didacticiel, vous allez AWS CodeBuild créer une collection d'exemples de fichiers d'entrée de code source (créer des artefacts d'entrée ou créer une entrée) dans une version déployable du code source (créer un artefact de sortie ou créer une sortie). Plus précisément, vous demandez d'CodeBuild utiliser Apache Maven, un outil de génération courant, pour créer un ensemble de fichiers de classe Java dans un fichier Java Archive (JAR). Vous n'avez pas besoin d'être familiarisé avec Apache Maven ni Java pour suivre ce didacticiel.

Vous pouvez travailler CodeBuild avec la CodeBuild console AWS CodePipeline, le AWS CLI, ou le AWS SDKs. Ce didacticiel explique comment utiliser la CodeBuild console. Pour obtenir des informations sur l'utilisation d' CodePipeline, veuillez consulter [À utiliser CodeBuild avec CodePipeline](#).

Important

Les étapes de ce didacticiel vous obligent à créer des ressources (par exemple, un compartiment S3) susceptibles d'entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [AWS CodeBuild les](#)

[sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatch Tarification Amazon](#).

Rubriques

- [Étape 1 : Création du code source](#)
- [Étape 2 : Création du fichier buildspec](#)
- [Étape 3 : Création de deux compartiments S3](#)
- [Étape 4 : Charger le code source et le fichier buildspec](#)
- [Étape 5 : Créer le projet de génération](#)
- [Étape 6 : Exécuter la génération](#)
- [Étape 7 : Afficher les informations de génération résumées](#)
- [Étape 8 : Afficher les informations de génération détaillées](#)
- [Étape 9 : Obtenir l'artefact de sortie de génération](#)
- [Étape 10 : Supprimer les compartiments S3](#)
- [Récapitulatif](#)

Étape 1 : Création du code source

(Partie de : [Commencer à AWS CodeBuild utiliser la console](#))

Au cours de cette étape, vous créez le code source que vous CodeBuild souhaitez intégrer au bucket de sortie. Ce code source se compose de deux fichiers de classe Java et d'un fichier POM (Project Object Model) Maven Apache.

1. Dans un répertoire vide sur votre ordinateur ou instance local, créez cette structure de répertoire.

```
(root directory name)
|-- src
|   |-- main
|       |-- java
|   |-- test
|       |-- java
```

2. A l'aide d'un éditeur de texte de votre choix, créez ce fichier, nommez-le `MessageUtil.java`, puis enregistrez-le dans le répertoire `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }

    public String salutationMessage() {
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

Ce fichier de classe crée en sortie la chaîne de caractères qui lui est transmise. Le constructeur `MessageUtil` définit la chaîne de caractères. La méthode `printMessage` crée la sortie. La méthode `salutationMessage` produit la sortie `Hi!` suivie par la chaîne de caractères.

3. Créez ce fichier, nommez-le `TestMessageUtil.java`, puis enregistrez-le dans le répertoire `/src/test/java`.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
```

```
public void testSalutationMessage() {
    System.out.println("Inside testSalutationMessage()");
    message = "Hi!" + "Robert";
    assertEquals(message,messageUtil.salutationMessage());
}
}
```

Ce fichier de classe définit la variable `message` dans la classe `MessageUtil` sur Robert. Il teste ensuite si la variable `message` a été correctement définie en vérifiant si les chaînes Robert et Hi!Robert apparaissent dans la sortie.

4. Créez ce fichier, nommez-le `pom.xml`, puis enregistrez-le dans le répertoire racine (de niveau supérieur).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>messageUtil</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
    <name>Message Utility Java Sample App</name>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
            </plugin>
        </plugins>
    </build>
</project>
```

Apache Maven utilise les instructions de ce fichier pour convertir les fichiers `MessageUtil.java` et `TestMessageUtil.java` en un fichier nommé `messageUtil-1.0.jar`, puis exécuter les tests spécifiés.

A ce stade, votre structure de répertoire ressemble à ceci.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Étape 2 : Création du fichier buildspec

(Étape précédente : [Étape 1 : Création du code source](#))

Au cours de cette étape, vous créez un fichier de spécification de génération (spécification de génération). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Sans spécification de construction, CodeBuild impossible de convertir avec succès votre entrée de génération en sortie de construction ou de localiser l'artefact de sortie de construction dans l'environnement de construction à télécharger dans votre compartiment de sortie.

Créez ce fichier, nommez-le `buildspec.yml`, puis enregistrez-le dans le répertoire racine (de niveau supérieur).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
```

```
build:
  commands:
    - echo Build started on `date`
    - mvn install
post_build:
  commands:
    - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Comme une déclaration de spécification de génération doit être à un format YAML valide, les espaces dans celle-ci sont importants. Si le nombre d'espaces dans votre déclaration de spécification de génération ne correspond pas, la génération peut échouer immédiatement. Vous pouvez utiliser un validateur YAML pour tester si votre déclaration de spécification de génération est à un format YAML valide.

Note

Au lieu d'inclure le fichier de spécification de génération dans votre code source, vous pouvez déclarer des commandes de génération séparément lorsque vous créez un projet de génération. Cela s'avère utile si vous souhaitez générer votre code source avec des commandes de génération différentes sans mettre à jour votre référentiel de code source chaque fois. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

Dans cette déclaration de spécification de génération :

- `version` représente la version de la norme de spécification de génération utilisée. Cette déclaration de spécification de génération utilise la dernière version, `0.2`.
- `phases` représente les phases de génération au cours desquelles vous pouvez demander à CodeBuild d'exécuter des commandes. Ces phases de génération sont répertoriées ici sous les noms `install`, `pre_build`, `build` et `post_build`. Vous ne pouvez pas modifier l'orthographe des noms des phases de génération, et vous ne pouvez pas en créer d'autres.

Dans cet exemple, pendant la build phase, CodeBuild exécute la `mvn install` commande. Cette commande demande à Apache Maven de compiler, tester et packager les fichiers de classe Java compilés dans un artefact de sortie de génération. A des fins d'exhaustivité, quelques echo commandes sont placées dans chaque phase de génération de cet exemple. Lorsque vous afficherez les informations détaillées de génération ultérieurement dans ce didacticiel, la sortie de ces commandes echo vous aidera à mieux comprendre comment CodeBuild exécute les commandes et dans quel ordre. (Même si toutes les phases de génération sont incluses dans cet exemple, vous n'êtes pas obligé d'inclure une phase de génération si vous n'avez pas l'intention de lancer des commandes au cours de cette phase.) Pour chaque phase de construction, CodeBuild exécute chaque commande spécifiée, une par une, dans l'ordre indiqué, du début à la fin.

- `artifacts` représente l'ensemble des artefacts de sortie de construction qui sont CodeBuild chargés dans le compartiment de sortie. `files` représente les fichiers à inclure dans la sortie de compilation. CodeBuild télécharge le `messageUtil-1.0.jar` fichier unique trouvé dans le répertoire `target` relatif de l'environnement de construction. Le nom de fichier `messageUtil-1.0.jar` et le nom du répertoire `target` sont basés sur la façon dont Apache Maven crée et stocke les artefacts de sortie de génération pour cet exemple uniquement. Dans vos générations, ces noms de fichier et ces répertoires sont différents.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

A ce stade, votre structure de répertoire ressemble à ceci.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Étape 3 : Création de deux compartiments S3

(Étape précédente : [Étape 2 : Création du fichier buildspec](#))

Vous pouvez utiliser un seul compartiment pour ce didacticiel, mais l'utilisation de deux compartiments permet de voir plus facilement la provenance de l'entrée de génération et la destination de la sortie de génération.

- L'un de ces compartiments (le compartiment d'entrée) stocke l'entrée de génération. Dans ce didacticiel, le nom de ce compartiment d'entrée est `codebuild-region-ID-account-ID-input-bucket`, où se trouvent *region-ID* la AWS région du compartiment et *account-ID* l'identifiant de votre AWS compte.
- L'autre compartiment (le compartiment de sortie) stocke la sortie de génération. Dans ce didacticiel, le nom de ce compartiment de sortie est `codebuild-region-ID-account-ID-output-bucket`.

Si vous avez choisi des noms différents pour ces compartiments, veillez à les utiliser tout au long de ce didacticiel.

Ces deux compartiments doivent se trouver dans la même AWS région que vos builds. Par exemple, si vous demandez d'CodeBuild exécuter un build dans la région USA Est (Ohio), ces compartiments doivent également se trouver dans la région USA Est (Ohio).

Pour de plus amples informations, veuillez consulter [Création d'un compartiment](#) dans le Guide de l'utilisateur d'Amazon Simple Storage Service.

Note

Bien qu'il prenne CodeBuild également en charge les entrées de build stockées dans CodeCommit les référentiels Bitbucket et Bitbucket, ce didacticiel ne vous explique pas comment les utiliser. GitHub Pour de plus amples informations, veuillez consulter [Planification d'une génération](#).

Étape 4 : Charger le code source et le fichier buildspec

(Étape précédente : [Étape 3 : Création de deux compartiments S3](#))

Au cours de cette étape, vous ajoutez le code source et le fichier de spécification de génération dans le compartiment d'entrée.

À l'aide de l'utilitaire zip de votre système d'exploitation, créez un fichier nommé `MessageUtil.zip` qui inclut `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` et `buildspec.yml`.

La structure de répertoire du fichier `MessageUtil.zip` doit ressembler à ceci.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |    |-- java
    |    |    |-- MessageUtil.java
    |-- test
    |    |-- java
    |    |    |-- TestMessageUtil.java
```

 Important

N'incluez pas le répertoire (*root directory name*), mais seulement les répertoires et fichiers du répertoire (*root directory name*).

Chargez le fichier `MessageUtil.zip` dans le compartiment d'entrée nommé `codebuild-region-ID-account-ID-input-bucket`.

 Important

Pour les référentiels Bitbucket et CodeCommit GitHub, par convention, vous devez stocker un fichier de spécification de construction nommé `buildspec.yml` à la racine (niveau supérieur) de chaque référentiel ou inclure la déclaration de spécification de construction dans le cadre de la définition du projet de construction. Ne créez pas un fichier ZIP qui contient le code source du référentiel et le fichier de spécification de génération.

Pour une entrée de génération stockée dans des compartiments S3 uniquement, vous devez créer un fichier ZIP contenant le code source et, par convention, un fichier de spécification de génération nommé `buildspec.yml` à la racine (niveau supérieur) ou inclure la déclaration de spécification de génération dans la définition de projet de génération.

Si vous souhaitez utiliser un autre nom pour votre fichier de spécification de génération ou que vous souhaitez créer une référence à une spécification de génération dans un emplacement autre que la racine, vous pouvez spécifier un remplacement de spécification de génération dans la définition de projet de génération. Pour de plus amples informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).

Étape 5 : Créer le projet de génération

(Étape précédente : [Étape 4 : Charger le code source et le fichier buildspec](#))

Au cours de cette étape, vous créez un projet de génération AWS CodeBuild qui permet d'exécuter la génération. Un projet de build inclut des informations sur la façon d'exécuter une compilation, notamment sur l'emplacement du code source, sur l'environnement de compilation à utiliser, sur les commandes de compilation à exécuter et sur l'emplacement où stocker le résultat de la compilation. Un environnement de génération représente une combinaison de système d'exploitation, d'environnement d'exécution d'un langage de programmation et d'outils CodeBuild utilisés pour exécuter une compilation. L'environnement de construction est exprimé sous forme d'image Docker. Pour de plus amples informations, veuillez consulter [Docker Overview](#) sur le site web Docker Docs.

Pour cet environnement de génération, vous demandez d' CodeBuild utiliser une image Docker contenant une version du kit de développement Java (JDK) et d'Apache Maven.

Pour créer le projet de génération

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Utilisez le sélecteur de AWS région pour choisir une AWS région prise en charge. CodeBuild Pour plus d'informations, consultez [Points de terminaison et quotas AWS CodeBuild](#) dans le document Référence générale d'Amazon Web Services.
3. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
4. Sur la page Create build project (Créer un projet de génération), dans Project configuration (Configuration du projet), pour Nom du projet, saisissez un nom pour ce projet de génération (dans cet exemple, codebuild-demo-project). Les noms des projets de construction doivent être uniques pour chaque AWS compte. Si vous utilisez un autre nom, veillez à l'utiliser tout au long de ce didacticiel.

Note

Sur la page Create build project (Créer un projet de génération), un message d'erreur similaire à ce qui suit peut s'afficher : You are not authorized to perform this operation (Vous n'êtes pas autorisé à effectuer cette opération). Cela est probablement dû au fait que vous vous êtes connecté au en AWS Management Console tant qu'utilisateur

qui n'est pas autorisé à créer un projet de construction. Pour résoudre ce problème, déconnectez-vous de AWS Management Console, puis reconnectez-vous avec les informations d'identification appartenant à l'une des entités IAM suivantes :

- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte dont les politiques `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, et `IAMFullAccess` gérées sont associées à cet utilisateur ou à un groupe IAM auquel appartient l'utilisateur. Si aucun utilisateur ou groupe de votre AWS compte ne possède ces autorisations et que vous ne pouvez pas les ajouter à votre utilisateur ou à votre groupe, contactez l'administrateur de votre AWS compte pour obtenir de l'aide. Pour de plus amples informations, veuillez consulter [AWS politiques gérées \(prédéfinies\) pour AWS CodeBuild](#).

Ces deux options incluent des autorisations d'administrateur qui vous permettent de créer un projet de génération afin que vous puissiez suivre ce didacticiel. Nous vous recommandons de toujours utiliser les autorisations minimales requises pour accomplir votre tâche. Pour de plus amples informations, veuillez consulter [AWS CodeBuild référence aux autorisations](#).

5. Dans Source, dans Source provider, choisissez Amazon S3.
6. Pour Bucket, choisissez `codebuild-region-ID - account-ID -input-bucket`.
7. Pour S3 object key (Clé d'objet S3), saisissez `MessageUtil.zip`.
8. Dans Environnement, pour Image d'environnement, laissez Image gérée sélectionné.
9. Pour Système d'exploitation, choisissez Amazon Linux.
10. Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
11. Pour Image, choisissez `aws/codebuild/amazonlinux-x86_64-standard:corretto11`.
12. Dans Rôle de service, laissez Nouveau rôle de service sélectionné et laissez Nom du rôle inchangé.
13. Pour Buildspec, laissez Utiliser un fichier buildspec sélectionné.
14. Dans Artifacts, pour Type, choisissez Amazon S3.
15. Pour le nom du compartiment, choisissez `codebuild-region-ID - account-ID -output-bucket`.

16. Laissez Nom et Chemin d'accès vides.
17. Choisissez Créer un projet de génération.

Étape 6 : Exécuter la génération

(Étape précédente : [Étape 5 : Créer le projet de génération](#))

Dans cette étape, vous demandez d' AWS CodeBuild exécuter le build avec les paramètres du projet de build.

Pour exécuter la génération

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de construction, choisissez codebuild-demo-project, puis choisissez Démarrer la construction. La construction démarre immédiatement.

Étape 7 : Afficher les informations de génération résumées

(Étape précédente : [Étape 6 : Exécuter la génération](#))

Au cours de cette étape, vous affichez un résumé de l'état de votre génération.

Pour afficher les informations de génération résumées

1. Si la **<build-ID>** page codebuild-demo-project: n'est pas affichée, dans la barre de navigation, choisissez Historique des constructions. Ensuite, dans la liste des projets de construction, pour Project, choisissez le lien Build run pour codebuild-demo-project. Il ne devrait y avoir qu'un seul lien correspondant. (Si vous avez déjà suivi ce didacticiel, choisissez le lien avec la valeur la plus récente dans la colonne Terminé.)
2. Sur la page État de la construction, dans Détails de la phase, les phases de construction suivantes doivent être affichées, avec la mention Succeeded dans la colonne État :
 - SUBMITTED
 - QUEUED
 - PROVISIONING

- DOWNLOAD_SOURCE
- INSTALL
- PRE_BUILD
- BUILD
- POST_BUILD
- UPLOAD_ARTIFACTS
- FINALIZING
- TERMINÉ

Dans État de la génération, Réussi doit être affiché.

Si vous voyez En cours à la place, choisissez le bouton d'actualisation.

3. En regard de chaque phase de génération, la valeur Durée indique combien de temps la phase de génération a duré. La valeur Heure de fin indique quand cette phase de génération s'est terminée.

Étape 8 : Afficher les informations de génération détaillées

(Étape précédente : [Étape 7 : Afficher les informations de génération résumées](#))

Au cours de cette étape, vous pouvez consulter des informations détaillées sur votre build dans CloudWatch Logs.

Note

Pour protéger les informations sensibles, les informations suivantes sont masquées dans CodeBuild les journaux :

- AWS clé d'accès IDs. Pour plus d'informations, consultez [la section Gestion des clés d'accès pour les utilisateurs IAM](#) dans le guide de l'Gestion des identités et des accès AWS utilisateur.
- Chaînes spécifiées à l'aide du stockage de paramètres. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

- Chaînes spécifiées à l'aide de AWS Secrets Manager. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

Pour afficher les informations de génération détaillées

1. Avec la page de détails de génération toujours affichée de l'étape précédente, les 10,000 dernières lignes du journal de génération apparaissent dans Journaux de génération. Pour voir l'intégralité du journal de construction dans CloudWatch Logs, cliquez sur le lien [Afficher le journal complet](#).
2. Dans le flux du journal CloudWatch des journaux, vous pouvez parcourir les événements du journal. Par défaut, seul le dernier ensemble d'événements de journal s'affiche. Pour afficher des événements de journal précédents, faites défiler vers la liste.
3. Dans ce didacticiel, la plupart des événements de journal contiennent des informations détaillées sur le téléchargement et l'installation des fichiers de dépendance de génération par CodeBuild dans son environnement de génération, ce qui ne vous intéresse probablement pas. Vous pouvez utiliser la zone Filtrer les événements pour limiter les informations affichées. Par exemple, si vous entrez "[INFO]" dans Filtrer les événements, seuls les événements contenant [INFO] s'affichent. Pour plus d'informations, consultez la section [Syntaxe des filtres et](#) des modèles dans le guide de CloudWatch l'utilisateur Amazon.

Étape 9 : Obtenir l'artefact de sortie de génération

(Étape précédente : [Étape 8 : Afficher les informations de génération détaillées](#))

Au cours de cette étape, vous obtenez le messageUtil-1.0.jar fichier qui CodeBuild a été créé et chargé dans le bucket de sortie.

Vous pouvez utiliser la CodeBuild console ou la console Amazon S3 pour effectuer cette étape.

Pour obtenir l'artefact de sortie de construction (AWS CodeBuild console)

1. La CodeBuild console étant toujours ouverte et la page des détails du build toujours affichée à l'étape précédente, choisissez l'onglet Détails du build et faites défiler la page vers le bas jusqu'à la section Artefacts.

 Note

Si la page des détails du build n'est pas affichée, dans la barre de navigation, choisissez Build history, puis cliquez sur le lien Build run.

2. Le lien vers le dossier Amazon S3 se trouve sous l'emplacement de téléchargement des artefacts. Ce lien ouvre le dossier dans Amazon S3 où se trouve le fichier d'artefact de sortie de messageUtil-1.0.jar compilation.

Pour obtenir l'artefact de sortie du build (console Amazon S3)

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Ouvrir codebuild-*region-ID-account-ID*-output-bucket.
3. Ouvrez le dossier codebuild-demo-project.
4. Ouvrez le dossier target, dans lequel se trouve le fichier d'artefact de sortie de génération messageUtil-1.0.jar.

Étape 10 : Supprimer les compartiments S3

(Étape précédente : [Étape 9 : Obtenir l'artefact de sortie de génération](#))

Pour éviter des frais permanents sur votre AWS compte, vous pouvez supprimer les compartiments d'entrée et de sortie utilisés dans ce didacticiel. Pour obtenir des instructions, consultez [Supprimer ou vider un compartiment](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Si vous utilisez l'utilisateur IAM ou un utilisateur administrateur IAM pour supprimer ces compartiments, l'utilisateur doit disposer d'autorisations d'accès supplémentaires. Ajoutez l'instruction suivante entre les marqueurs (**### BEGIN ADDING STATEMENT HERE ###** et **### END ADDING STATEMENTS HERE ###**) à une politique d'accès existante pour l'utilisateur.

Les points de suspension (...) figurant dans cette instruction sont utilisés par souci de concision. Ne supprimez aucune instruction de la stratégie d'accès existante. Ne saisissez pas ces ellipses dans la stratégie.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Récapitulatif

Dans ce didacticiel, vous avez AWS CodeBuild créé un ensemble de fichiers de classe Java dans un fichier JAR. Vous avez ensuite affiché les résultats de la génération.

Vous pouvez maintenant essayer de l'utiliser CodeBuild dans vos propres scénarios. Suivez les instructions de la section [Planification d'une génération](#). Si vous ne vous sentez pas encore fait prêt, vous pouvez essayer de générer certains des exemples. Pour de plus amples informations, veuillez consulter [Utilisez des exemples basés sur des cas pour CodeBuild](#).

Commencer à AWS CodeBuild utiliser le AWS CLI

Dans ce didacticiel, vous allez AWS CodeBuild créer une collection d'exemples de fichiers d'entrée de code source (appelés build input artefacts ou build input) dans une version déployable du code source (appelée build output artefact ou build output). Plus précisément, vous demandez d'CodeBuild utiliser Apache Maven, un outil de génération courant, pour créer un ensemble de fichiers de classe Java dans un fichier Java Archive (JAR). Vous n'avez pas besoin d'être familiarisé avec Apache Maven ni Java pour suivre ce didacticiel.

Vous pouvez travailler CodeBuild avec la CodeBuild console AWS CodePipeline, le AWS CLI, ou le AWS SDKs. Ce didacticiel explique comment utiliser CodeBuild le AWS CLI. Pour plus d'informations sur l'utilisation CodePipeline, consultez [À utiliser CodeBuild avec CodePipeline](#).

Important

Les étapes de ce didacticiel vous obligent à créer des ressources (par exemple, un compartiment S3) susceptibles d'entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [CodeBuildles sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatch Tarification Amazon](#).

Rubriques

- [Étape 1 : Création du code source](#)
- [Étape 2 : Création du fichier buildspec](#)
- [Étape 3 : Création de deux compartiments S3](#)
- [Étape 4 : Charger le code source et le fichier buildspec](#)
- [Étape 5 : Créer le projet de génération](#)
- [Étape 6 : Exécuter la génération](#)
- [Étape 7 : Afficher les informations de génération résumées](#)
- [Étape 8 : Afficher les informations de génération détaillées](#)
- [Étape 9 : Obtenir l'artefact de sortie de génération](#)
- [Étape 10 : Supprimer les compartiments S3](#)
- [Récapitulatif](#)

Étape 1 : Création du code source

(Partie de : [Commencer à AWS CodeBuild utiliser le AWS CLI](#))

Au cours de cette étape, vous créez le code source que vous CodeBuild souhaitez intégrer au bucket de sortie. Ce code source se compose de deux fichiers de classe Java et d'un fichier POM (Project Object Model) Maven Apache.

1. Dans un répertoire vide sur votre ordinateur ou instance local, créez cette structure de répertoire.

```
(root directory name)
|-- src
```

```
|-- main
|   |-- java
|-- test
|   |-- java
```

2. A l'aide d'un éditeur de texte de votre choix, créez ce fichier, nommez-le `MessageUtil.java`, puis enregistrez-le dans le répertoire `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }

    public String salutationMessage() {
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

Ce fichier de classe crée en sortie la chaîne de caractères qui lui est transmise. Le constructeur `MessageUtil` définit la chaîne de caractères. La méthode `printMessage` crée la sortie. La méthode `salutationMessage` produit la sortie `Hi!` suivie par la chaîne de caractères.

3. Créez ce fichier, nommez-le `TestMessageUtil.java`, puis enregistrez-le dans le répertoire `/src/test/java`.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);
```

```
@Test
public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    assertEquals(message,messageUtil.printMessage());
}

@Test
public void testSalutationMessage() {
    System.out.println("Inside testSalutationMessage()");
    message = "Hi!" + "Robert";
    assertEquals(message,messageUtil.salutationMessage());
}
}
```

Ce fichier de classe définit la variable `message` dans la classe `MessageUtil` sur Robert. Il teste ensuite si la variable `message` a été correctement définie en vérifiant si les chaînes Robert et Hi!Robert apparaissent dans la sortie.

4. Créez ce fichier, nommez-le `pom.xml`, puis enregistrez-le dans le répertoire racine (de niveau supérieur).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>messageUtil</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
    <name>Message Utility Java Sample App</name>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
```

```
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.0</version>
</plugin>
</plugins>
</build>
</project>
```

Apache Maven utilise les instructions de ce fichier pour convertir les fichiers `MessageUtil.java` et `TestMessageUtil.java` en un fichier nommé `messageUtil-1.0.jar`, puis exécuter les tests spécifiés.

A ce stade, votre structure de répertoire ressemble à ceci.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |    |-- java
    |    |    |-- MessageUtil.java
    |-- test
    |    |-- java
    |    |    |-- TestMessageUtil.java
```

Étape 2 : Création du fichier buildspec

(Étape précédente : [Étape 1 : Création du code source](#))

Au cours de cette étape, vous créez un fichier de spécification de génération (spécification de génération). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Sans spécification de construction, CodeBuild impossible de convertir avec succès votre entrée de génération en sortie de construction ou de localiser l'artefact de sortie de construction dans l'environnement de construction à télécharger dans votre compartiment de sortie.

Créez ce fichier, nommez-le `buildspec.yml`, puis enregistrez-le dans le répertoire racine (de niveau supérieur).

```
version: 0.2

phases:
```

```
install:
  runtime-versions:
    java: corretto11
pre_build:
  commands:
    - echo Nothing to do in the pre_build phase...
build:
  commands:
    - echo Build started on `date`
    - mvn install
post_build:
  commands:
    - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Comme une déclaration de spécification de génération doit être à un format YAML valide, les espaces dans celle-ci sont importants. Si le nombre d'espaces dans votre déclaration de spécification de génération ne correspond pas, la génération peut échouer immédiatement. Vous pouvez utiliser un validateur YAML pour tester si votre déclaration de spécification de génération est à un format YAML valide.

Note

Au lieu d'inclure le fichier de spécification de génération dans votre code source, vous pouvez déclarer des commandes de génération séparément lorsque vous créez un projet de génération. Cela s'avère utile si vous souhaitez générer votre code source avec des commandes de génération différentes sans mettre à jour votre référentiel de code source chaque fois. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

Dans cette déclaration de spécification de génération :

- `version` représente la version de la norme de spécification de génération utilisée. Cette déclaration de spécification de génération utilise la dernière version, `0.2`.

- `phases` représente les phases de génération au cours desquelles vous pouvez demander à CodeBuild d'exécuter des commandes. Ces phases de génération sont répertoriées ici sous les noms `install`, `pre_build`, `build` et `post_build`. Vous ne pouvez pas modifier l'orthographe des noms des phases de génération, et vous ne pouvez pas en créer d'autres.

Dans cet exemple, pendant la `build` phase, CodeBuild exécute la `mvn install` commande. Cette commande demande à Apache Maven de compiler, tester et packager les fichiers de classe Java compilés dans un artefact de sortie de génération. A des fins d'exhaustivité, quelques `echo` commandes sont placées dans chaque phase de génération de cet exemple. Lorsque vous afficherez les informations détaillées de génération ultérieurement dans ce didacticiel, la sortie de ces commandes `echo` vous aidera à mieux comprendre comment CodeBuild exécute les commandes et dans quel ordre. (Même si toutes les phases de génération sont incluses dans cet exemple, vous n'êtes pas obligé d'inclure une phase de génération si vous n'avez pas l'intention de lancer des commandes au cours de cette phase.) Pour chaque phase de construction, CodeBuild exécute chaque commande spécifiée, une par une, dans l'ordre indiqué, du début à la fin.

- `artifacts` représente l'ensemble des artefacts de sortie de construction qui sont CodeBuild chargés dans le compartiment de sortie. `files` représente les fichiers à inclure dans la sortie de compilation. CodeBuild télécharge le `messageUtil-1.0.jar` fichier unique trouvé dans le répertoire `target` relatif de l'environnement de construction. Le nom de fichier `messageUtil-1.0.jar` et le nom du répertoire `target` sont basés sur la façon dont Apache Maven crée et stocke les artefacts de sortie de génération pour cet exemple uniquement. Dans vos générations, ces noms de fichier et ces répertoires sont différents.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

A ce stade, votre structure de répertoire ressemble à ceci.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |    |-- java
    |    |    |-- MessageUtil.java
    |-- test
    |    |-- java
    |    |    |-- TestMessageUtil.java
```

Étape 3 : Création de deux compartiments S3

(Étape précédente : [Étape 2 : Création du fichier buildspec](#))

Vous pouvez utiliser un seul compartiment pour ce didacticiel, mais l'utilisation de deux compartiments permet de voir plus facilement la provenance de l'entrée de génération et la destination de la sortie de génération.

- L'un de ces compartiments (le compartiment d'entrée) stocke l'entrée de génération. Dans ce didacticiel, le nom de ce compartiment d'entrée est `codebuild-region-ID-account-ID-input-bucket`, où se trouvent *region-ID* la AWS région du compartiment et *account-ID* l'identifiant de votre AWS compte.
- L'autre compartiment (le compartiment de sortie) stocke la sortie de génération. Dans ce didacticiel, le nom de ce compartiment de sortie est `codebuild-region-ID-account-ID-output-bucket`.

Si vous avez choisi des noms différents pour ces compartiments, veillez à les utiliser tout au long de ce didacticiel.

Ces deux compartiments doivent se trouver dans la même AWS région que vos builds. Par exemple, si vous demandez d' CodeBuild exécuter un build dans la région USA Est (Ohio), ces compartiments doivent également se trouver dans la région USA Est (Ohio).

Pour de plus amples informations, veuillez consulter [Création d'un compartiment](#) dans le Guide de l'utilisateur d'Amazon Simple Storage Service.

Note

Bien qu'il prenne CodeBuild également en charge les entrées de build stockées dans CodeCommit les référentiels Bitbucket et Bitbucket, ce didacticiel ne vous explique pas comment les utiliser. GitHub Pour de plus amples informations, veuillez consulter [Planification d'une génération](#).

Étape 4 : Charger le code source et le fichier buildspec

(Étape précédente : [Étape 3 : Création de deux compartiments S3](#))

Au cours de cette étape, vous ajoutez le code source et le fichier de spécification de génération dans le compartiment d'entrée.

À l'aide de l'utilitaire zip de votre système d'exploitation, créez un fichier nommé `MessageUtil.zip` qui inclut `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` et `buildspec.yml`.

La structure de répertoire du fichier `MessageUtil.zip` doit ressembler à ceci.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Important

N'incluez pas le répertoire (*root directory name*), mais seulement les répertoires et fichiers du répertoire (*root directory name*).

Chargez le fichier `MessageUtil.zip` dans le compartiment d'entrée nommé `codebuild-region-ID-account-ID-input-bucket`.

Important

Pour les référentiels Bitbucket et CodeCommit GitHub, par convention, vous devez stocker un fichier de spécification de construction nommé `buildspec.yml` à la racine (niveau supérieur) de chaque référentiel ou inclure la déclaration de spécification de construction dans le cadre de la définition du projet de construction. Ne créez pas un fichier ZIP qui contient le code source du référentiel et le fichier de spécification de génération.

Pour une entrée de génération stockée dans des compartiments S3 uniquement, vous devez créer un fichier ZIP contenant le code source et, par convention, un fichier de spécification de génération nommé `buildspec.yml` à la racine (niveau supérieur) ou inclure la déclaration de spécification de génération dans la définition de projet de génération.

Si vous souhaitez utiliser un autre nom pour votre fichier de spécification de génération ou que vous souhaitez créer une référence à une spécification de génération dans un emplacement autre que la racine, vous pouvez spécifier un remplacement de spécification de génération dans la définition de projet de génération. Pour de plus amples informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).

Étape 5 : Créer le projet de génération

(Étape précédente : [Étape 4 : Charger le code source et le fichier buildspec](#))

Au cours de cette étape, vous créez un projet de génération AWS CodeBuild qui permet d'exécuter la génération. Un projet de build inclut des informations sur la façon d'exécuter une compilation, notamment sur l'emplacement du code source, sur l'environnement de compilation à utiliser, sur les commandes de compilation à exécuter et sur l'emplacement où stocker le résultat de la compilation. Un environnement de génération représente une combinaison de système d'exploitation, d'environnement d'exécution d'un langage de programmation et d'outils CodeBuild utilisés pour exécuter une compilation. L'environnement de construction est exprimé sous forme d'image Docker. Pour de plus amples informations, veuillez consulter [Docker Overview](#) sur le site web Docker Docs.

Pour cet environnement de génération, vous demandez d' CodeBuild utiliser une image Docker contenant une version du kit de développement Java (JDK) et d'Apache Maven.

Pour créer le projet de génération

1. Utilisez le AWS CLI pour exécuter la create-project commande :

```
aws codebuild create-project --generate-cli-skeleton
```

Des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier nommé `create-project.json` à un emplacement de l'ordinateur local ou de l'instance où le AWS CLI est installé. Si vous choisissez d'utiliser un autre nom de fichier, veillez à l'utiliser tout au long de ce didacticiel.

Modifiez les données copiées selon le format suivant, puis enregistrez vos résultats :

```
{
  "name": "codebuild-demo-project",
  "source": {
```

```
{
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"artifacts": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-output-bucket"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "serviceIAMRole"
}
```

serviceIAMRole Remplacez-le par le nom de ressource Amazon (ARN) d'un rôle de CodeBuild service (par exemple, `arn:aws:iam::account-ID:role/role-name`). Pour en créer un, consultez [CodeBuild Autoriser l'interaction avec d'autres AWS services](#).

Dans ces données :

- `name` représente un identificateur obligatoire pour ce projet de génération (dans cet exemple, `codebuild-demo-project`). Les noms de projet de génération doivent être uniques au sein de votre compte.
- Pour `source`, `type` est une valeur obligatoire qui représente le type de référentiel du code source (dans cet exemple, `S3` pour un compartiment Amazon S3).
- Pour `source`, `location` représente le chemin d'accès au code source (dans cet exemple, le nom du compartiment d'entrée, suivi du nom du fichier ZIP).
- Pour `artifacts`, `type` est une valeur obligatoire qui représente le type de référentiel de l'artefact de sortie de construction (dans cet exemple, `S3` pour un compartiment Amazon S3).
- Pour `artifacts`, `location` représente le nom du compartiment de sortie que vous avez créé ou identifié précédemment (dans cet exemple, `codebuild-region-ID-account-ID-output-bucket`).
- Pour `environment`, `type` est une valeur obligatoire qui représente le type d'environnement de construction (dans cet exemple, `LINUX_CONTAINER`).
- Pour `environment`, `image` est une valeur obligatoire qui représente le nom de l'image Docker et la combinaison de balises utilisés par ce projet de génération, telle que spécifiée par le type de référentiel d'images Docker (dans cet exemple, `aws/codebuild/standard:5.0`).

pour une image Docker dans le référentiel d'images CodeBuild Docker). `aws/codebuild/standard` est le nom de l'image Docker. `5.0` est le tag de l'image Docker.

Pour trouver d'autres images Docker que vous pouvez utiliser dans vos scénarios, consultez [Référence sur les environnements de génération](#).

- Pour `environment`, `computeType` est une valeur obligatoire qui représente les ressources informatiques CodeBuild utilisées (dans cet exemple, `BUILD_GENERAL1_SMALL`).

Note

Les autres valeurs disponibles dans les données au format JSON d'origine, comme `description`, `buildspec`, `auth` (y compris `type` et `resource`), `path`, `namespaceType`, `name` (pour `artifacts`), `packaging`, `environmentVariables` (y compris `name` et `value`), `timeoutInMinutes`, `encryptionKey` et `tags` (y compris `key` et `value`) sont facultatives. Elles ne sont pas utilisées dans ce didacticiel et ne sont donc pas montrées ici. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

2. Placez-vous dans le répertoire contenant le fichier que vous venez d'enregistrer, puis exécutez à nouveau la commande `create-project`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

En cas de réussite, des données similaires à celles-ci apparaissent dans la sortie.

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
  }
}
```

```
"environment": {
  "computeType": "BUILD_GENERAL1_SMALL",
  "image": "aws/codebuild/standard:5.0",
  "type": "LINUX_CONTAINER",
  "environmentVariables": []
},
"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
"arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
}
```

- **project** représente les informations sur ce projet de génération.
- **tags** représente les balises qui ont été déclarées.
- **packaging** représente la façon dont l'artefact de sortie de génération est stocké dans le compartiment de sortie. **NONE** signifie qu'un dossier est créé dans le compartiment de sortie. L'artefact de sortie de génération est stocké dans ce dossier.
- **lastModified** représente l'heure, au format horaire Unix, à laquelle les informations sur le projet de génération ont été modifiées pour la dernière fois.
- **timeoutInMinutes** représente le nombre de minutes après lesquelles la génération CodeBuild s'arrête si la génération n'est pas terminée. (La valeur par défaut est de 60 minutes.)
- **created** représente l'heure, au format horaire Unix, à laquelle le projet de génération a été créé.
- **environmentVariables** représente toutes les variables d'environnement qui ont été déclarées et qui CodeBuild peuvent être utilisées pendant la construction.
- **encryptionKey** représente l'ARN de la clé gérée par le client qui a été CodeBuild utilisée pour chiffrer l'artefact de sortie de construction.
- **arn** représente l'ARN du projet de génération.

Note

Après avoir exécuté la `create-project` commande, un message d'erreur similaire au suivant peut s'afficher : Utilisateur : n'***user-ARN***est pas autorisé à exécuter : codebuild :

CreateProject. Cela est probablement dû au fait que vous avez configuré le AWS CLI avec les informations d'identification d'un utilisateur qui ne dispose pas des autorisations suffisantes CodeBuild pour créer des projets de construction. Pour résoudre ce problème, configurez les informations AWS CLI d'identification appartenant à l'une des entités IAM suivantes :

- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte dont les politiques `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, et `IAMFullAccess` gérées sont associées à cet utilisateur ou à un groupe IAM auquel appartient l'utilisateur. Si aucun utilisateur ou groupe de votre AWS compte ne possède ces autorisations et que vous ne pouvez pas les ajouter à votre utilisateur ou à votre groupe, contactez l'administrateur de votre AWS compte pour obtenir de l'aide. Pour de plus amples informations, veuillez consulter [AWS politiques gérées \(prédéfinies\) pour AWS CodeBuild](#).

Étape 6 : Exécuter la génération

(Étape précédente : [Étape 5 : Créer le projet de génération](#))

Dans cette étape, vous demandez d' AWS CodeBuild exécuter le build avec les paramètres du projet de build.

Pour exécuter la génération

1. Utilisez le AWS CLI pour exécuter la start-build commande :

```
aws codebuild start-build --project-name project-name
```

project-name Remplacez-le par le nom de votre projet de construction de l'étape précédente (par exemple, `codebuild-demo-project`).

2. En cas de réussite, des données similaires à ce qui suit s'affichent dans la sortie :

```
{
  "build": {
    "buildComplete": false,
```

```
"initiator": "user-name",
"artifacts": {
  "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/
message-util.zip"
},
"projectName": "codebuild-demo-project",
"timeoutInMinutes": 60,
"buildStatus": "IN_PROGRESS",
"environment": {
  "computeType": "BUILD_GENERAL1_SMALL",
  "image": "aws/codebuild/standard:5.0",
  "type": "LINUX_CONTAINER",
  "environmentVariables": []
},
"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"currentPhase": "SUBMITTED",
"startTime": 1472848787.882,
"id": "codebuild-demo-project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE",
"arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE"
}
}
```

- build représente les informations sur cette génération.
- buildComplete indique si la génération a été terminée (true). Sinon la valeur est renvoyé, false.
- initiator représente l'entité qui a démarré la génération.
- artifacts représente les informations sur la sortie de génération, y compris son emplacement.
- projectName représente le nom du projet de génération.
- buildStatus représente l'état actuel de la génération lorsque la commande start-build a été exécutée.
- currentPhase représente la phase actuelle de génération lorsque la commande start-build a été exécutée.
- startTime représente l'heure, au format horaire Unix, à laquelle le processus de génération a démarré.

- `id` représente l'ID de la génération.
- `arn` représente l'ARN de la génération.

Notez la valeur `id`. Vous en avez besoin à l'étape suivante.

Étape 7 : Afficher les informations de génération résumées

(Étape précédente : [Étape 6 : Exécuter la génération](#))

Au cours de cette étape, vous affichez un résumé de l'état de votre génération.

Pour afficher les informations de génération résumées

- Utilisez le AWS CLI pour exécuter la `batch-get-builds` commande.

```
aws codebuild batch-get-builds --ids id
```

Remplacez *id* par la `id` valeur qui apparaissait dans le résultat de l'étape précédente.

En cas de réussite, des données similaires à celles-ci apparaissent dans la sortie.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
          "startTime": 1472848787.882
        },
        ... The full list of build phases has been omitted for brevity ...
        {
          "phaseType": "COMPLETED",
          "startTime": 1472848878.079
        }
      ],
    }
  ],
}
```

```

    "logs": {
      "groupName": "/aws/codebuild/codebuild-demo-project",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
      "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
    },
    "artifacts": {
      "md5sum": "MD5-hash",
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/message-util.zip",
      "sha256sum": "SHA-256-hash"
    },
    "projectName": "codebuild-demo-project",
    "timeoutInMinutes": 60,
    "initiator": "user-name",
    "buildStatus": "SUCCEEDED",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "COMPLETED",
    "startTime": 1472848787.882,
    "endTime": 1472848878.079,
    "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
  }
]
}

```

- `buildsNotFound` représente la version IDs pour toutes les versions pour lesquelles aucune information n'est disponible. Dans cet exemple, cette valeur doit être vide.
- `builds` représente les informations pour chaque génération où les informations sont disponibles. Dans cet exemple, des informations sur une génération apparaissent dans la sortie.

- `phases` représente l'ensemble de phases de génération qu' CodeBuild exécute au cours du processus de génération. Les informations de chaque phase de génération sont répertoriées séparément sous la forme `startTime`, `endTime`, `durationInSeconds` (heures de début et de fin de la phase de génération au format horaire Unix et durée de la phase en secondes) et `phaseType`, par exemple, (SUBMITTED, PROVISIONING, DOWNLOAD_SOURCE, INSTALL, PRE_BUILD, BUILD, POST_BUILD, UPLOAD_ARTIFACTS, FINALIZING ou COMPLETED) et `phaseStatus` (par exemple, SUCCEEDED, FAILED, FAULT, TIMED_OUT, IN_PROGRESS ou STOPPED). La première fois que vous exécutez la commande `batch-get-builds`, il se peut qu'il n'y ait que peu (ou aucune) phases. Après les exécutions suivantes de la commande `batch-get-builds` avec le même ID de génération, plus de phases de génération devraient apparaître dans la sortie.
- `logs` représente les informations contenues dans Amazon CloudWatch Logs concernant les journaux du build.
- `md5sum` MD5 et `sha256sum` représentent les hachages SHA-256 de l'artefact de sortie de la version. Ces hachages s'affichent dans la sortie uniquement si la valeur `packaging` du projet de génération est définie sur ZIP. (Vous n'avez pas défini cette valeur dans ce didacticiel.) Vous pouvez utiliser ces hachages avec un outil de contrôle de somme pour vérifier l'intégrité et l'authenticité des fichiers.

 Note

Vous pouvez également utiliser la console Amazon S3 pour afficher ces hachages. Cochez la case en regard de l'artefact de sortie de génération, choisissez Actions, puis choisissez Propriétés. Dans le volet Propriétés, développez Metadata et visualisez les valeurs de `x-amz-meta-codebuild-content-md5` et `-content-sha256`. `x-amz-meta-codebuild` (Dans la console Amazon S3, la ETag valeur de l'artefact de sortie de compilation ne doit pas être interprétée comme étant le hachage MD5 SHA-256.)

Si vous utilisez le AWS SDKs pour obtenir ces hachages, les valeurs sont nommées `codebuild-content-md5` et `codebuild-content-sha256`.

- `endTime` représente l'heure, au format horaire Unix, à laquelle processus de génération s'est terminé.

 Note

Les métadonnées Amazon S3 ont un CodeBuild en-tête nommé `x-amz-meta-codebuild-buildarn` qui contient le nom `buildArn` de la CodeBuild version qui publie les artefacts sur Amazon S3. Le `buildArn` est ajouté pour permettre le suivi de la source des notifications et pour indiquer la version à partir de laquelle l'artefact est généré.

Étape 8 : Afficher les informations de génération détaillées

(Étape précédente : [Étape 7 : Afficher les informations de génération résumées](#))

Au cours de cette étape, vous pouvez consulter des informations détaillées sur votre build dans CloudWatch Logs.

 Note

Pour protéger les informations sensibles, les informations suivantes sont masquées dans CodeBuild les journaux :

- AWS clé d'accès IDs. Pour plus d'informations, consultez [la section Gestion des clés d'accès pour les utilisateurs IAM](#) dans le guide de l'Gestion des identités et des accès AWS utilisateur.
- Chaînes spécifiées à l'aide du stockage de paramètres. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.
- Chaînes spécifiées à l'aide de AWS Secrets Manager. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

Pour afficher les informations de génération détaillées

1. Utilisez votre navigateur Web pour accéder à l'emplacement `deepLink` qui apparaît dans la sortie de l'étape précédente (par exemple, `https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).

2. Dans le flux du journal CloudWatch des journaux, vous pouvez parcourir les événements du journal. Par défaut, seul le dernier ensemble d'événements de journal s'affiche. Pour afficher des événements de journal précédents, faites défiler vers la liste.
3. Dans ce didacticiel, la plupart des événements de journal contiennent des informations détaillées sur le téléchargement et l'installation des fichiers de dépendance de génération par CodeBuild dans son environnement de génération, ce qui ne vous intéresse probablement pas. Vous pouvez utiliser la zone Filtrer les événements pour limiter les informations affichées. Par exemple, si vous entrez "[INFO]" dans Filtrer les événements, seuls les événements contenant [INFO] s'affichent. Pour plus d'informations, consultez la section [Syntaxe des filtres et](#) des modèles dans le guide de CloudWatch l'utilisateur Amazon.

Ces parties d'un flux de CloudWatch log Logs se rapportent à ce didacticiel.

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
[Container] 2016/04/15 17:49:44 [INFO]
[Container] 2016/04/15 17:49:44 [INFO]
-----
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
-----
...
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55  T E S T S
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 Running TestMessageUtil
[Container] 2016/04/15 17:49:55 Inside testSalutationMessage()
[Container] 2016/04/15 17:49:55 Hi!Robert
[Container] 2016/04/15 17:49:55 Inside testPrintMessage()
[Container] 2016/04/15 17:49:55 Robert
```

```
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.018 sec
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Results :
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
...
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] BUILD SUCCESS
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] Total time: 11.845 s
[Container] 2016/04/15 17:49:56 [INFO] Finished at: 2016-04-15T17:49:56+00:00
[Container] 2016/04/15 17:49:56 [INFO] Final Memory: 18M/216M
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 Phase complete: BUILD Success: true
[Container] 2016/04/15 17:49:56 Entering phase POST_BUILD
[Container] 2016/04/15 17:49:56 Running command echo Entering post_build phase...
[Container] 2016/04/15 17:49:56 Entering post_build phase...
[Container] 2016/04/15 17:49:56 Phase complete: POST_BUILD Success: true
[Container] 2016/04/15 17:49:57 Preparing to copy artifacts
[Container] 2016/04/15 17:49:57 Assembling file list
[Container] 2016/04/15 17:49:57 Expanding target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Found target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Creating zip artifact
```

Dans cet exemple, les phases de pré-construction, de construction et de post-construction ont été terminées CodeBuild avec succès. Il a exécuté les tests unitaires et généré avec succès le fichier `messageUtil-1.0.jar`.

Étape 9 : Obtenir l'artefact de sortie de génération

(Étape précédente : [Étape 8 : Afficher les informations de génération détaillées](#))

Au cours de cette étape, vous obtenez le `messageUtil-1.0.jar` fichier qui CodeBuild a été créé et chargé dans le bucket de sortie.

Vous pouvez utiliser la CodeBuild console ou la console Amazon S3 pour effectuer cette étape.

Pour obtenir l'artefact de sortie de construction (AWS CodeBuild console)

1. La CodeBuild console étant toujours ouverte et la page des détails du build toujours affichée à l'étape précédente, choisissez l'onglet Détails du build et faites défiler la page vers le bas jusqu'à la section Artefacts.

 Note

Si la page des détails du build n'est pas affichée, dans la barre de navigation, choisissez Build history, puis cliquez sur le lien Build run.

2. Le lien vers le dossier Amazon S3 se trouve sous l'emplacement de téléchargement des artefacts. Ce lien ouvre le dossier dans Amazon S3 où se trouve le fichier d'artefact de sortie de messageUtil-1.0.jar compilation.

Pour obtenir l'artefact de sortie du build (console Amazon S3)

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Ouvrir codebuild-*region-ID-account-ID*-output-bucket.
3. Ouvrez le dossier codebuild-demo-project.
4. Ouvrez le dossier target, dans lequel se trouve le fichier d'artefact de sortie de génération messageUtil-1.0.jar.

Étape 10 : Supprimer les compartiments S3

(Étape précédente : [Étape 9 : Obtenir l'artefact de sortie de génération](#))

Pour éviter des frais permanents sur votre AWS compte, vous pouvez supprimer les compartiments d'entrée et de sortie utilisés dans ce didacticiel. Pour obtenir des instructions, consultez [Supprimer ou vider un compartiment](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Si vous utilisez l'utilisateur IAM ou un utilisateur administrateur IAM pour supprimer ces compartiments, l'utilisateur doit disposer d'autorisations d'accès supplémentaires. Ajoutez l'instruction suivante entre les marqueurs (**### BEGIN ADDING STATEMENT HERE ###** et **### END ADDING STATEMENTS HERE ###**) à une politique d'accès existante pour l'utilisateur.

Les points de suspension (...) figurant dans cette instruction sont utilisés par souci de concision. Ne supprimez aucune instruction de la stratégie d'accès existante. Ne saisissez pas ces ellipses dans la stratégie.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Récapitulatif

Dans ce didacticiel, vous avez AWS CodeBuild créé un ensemble de fichiers de classe Java dans un fichier JAR. Vous avez ensuite affiché les résultats de la génération.

Vous pouvez maintenant essayer de l'utiliser CodeBuild dans vos propres scénarios. Suivez les instructions de la section [Planification d'une génération](#). Si vous ne vous sentez pas encore fait prêt, vous pouvez essayer de générer certains des exemples. Pour de plus amples informations, veuillez consulter [Utilisez des exemples basés sur des cas pour CodeBuild](#).

Utilisez des exemples basés sur des cas pour CodeBuild

Vous pouvez utiliser ces exemples basés sur des cas pour expérimenter avec AWS CodeBuild :

[Échantillons multiservices](#)

Une liste d'échantillons multiservices à tester. AWS CodeBuild

[Exemple de badges de génération](#)

Montre comment procéder à la configuration CodeBuild avec des badges de construction.

[Exemple de rapport de test](#)

Utilise le AWS CLI pour créer, exécuter et afficher les résultats d'un rapport de test.

[Exemples Docker pour CodeBuild](#)

Montre comment utiliser des images Docker personnalisées, publier des images Docker dans un référentiel Amazon ECR et utiliser des images Docker dans un registre privé.

[Hébergement d'une sortie de génération dans un compartiment S3](#)

Montre comment créer un site web statique dans un compartiment S3 à l'aide d'artefacts de génération non chiffrés.

[Échantillon d'entrées et de sorties multiples](#)

Montre comment utiliser plusieurs sources d'entrée et plusieurs artefacts de sortie dans un projet de génération.

[Exemples d'exécution de tests parallèles](#)

Montre comment utiliser la commande `codebuild-tests-run` CLI pour diviser et exécuter des tests dans des environnements d'exécution parallèle.

[Versions d'environnement d'exécution dans l'exemple de fichier buildspec](#)

Montre comment spécifier des environnements d'exécution et leurs versions dans le fichier buildspec.

[Exemple de version source](#)

Montre comment utiliser une version spécifique de votre source dans un projet de CodeBuild génération.

[Exemples de référentiels sources tiers pour CodeBuild](#)

Montre comment créer BitBucket, GitHub Enterprise Server et GitHub extraire des requêtes à l'aide de webhooks en utilisant CodeBuild.

[Définissez les noms des artefacts au moment de la construction à l'aide du versionnement sémantique](#)

Montre comment utiliser la gestion sémantique des versions pour créer un nom d'artefact au moment de la génération.

Exemples multiservices pour CodeBuild

Vous pouvez utiliser ces exemples multiservices pour tester : AWS CodeBuild

[Exemple Amazon ECR](#)

Utilise une image Docker dans un référentiel Amazon ECR pour utiliser Apache Maven afin de produire un seul fichier JAR. Les exemples d'instructions vous montreront comment créer et envoyer une image Docker vers Amazon ECR, créer un projet Go, créer le projet, exécuter le projet et configurer les autorisations permettant de se connecter CodeBuild à Amazon ECR.

[Exemple Amazon EFS](#)

Montre comment configurer un fichier builds spec afin qu'un CodeBuild projet soit monté et construit sur un système de fichiers Amazon EFS. Les exemples d'instructions vous montreront comment créer un Amazon VPC, créer un système de fichiers dans Amazon VPC, créer et construire un projet utilisant Amazon VPC, puis passer en revue le fichier de projet et les variables générés.

[AWS CodePipeline échantillons](#)

Montre comment AWS CodePipeline créer un build avec des builds par lots ainsi que plusieurs sources d'entrée et plusieurs artefacts de sortie. Cette section contient des exemples de fichiers JSON qui montrent des structures de pipeline qui créent des constructions par lots avec des artefacts distincts et des artefacts combinés. Un exemple JSON supplémentaire est fourni qui montre la structure du pipeline avec plusieurs sources d'entrée et plusieurs artefacts de sortie.

[AWS Config échantillon](#)

Montre comment procéder à la configuration AWS Config. Répertorie les CodeBuild ressources qui font l'objet d'un suivi et décrit comment rechercher CodeBuild des projets AWS Config. Les exemples d'instructions vous indiqueront les conditions préalables à l'intégration AWS Config,

les étapes de configuration AWS Config et les étapes de recherche de CodeBuild projets et de données. AWS Config

[Exemple de notifications de génération](#)

Utilise Apache Maven pour produire un fichier JAR unique. Envoie une notification de création aux abonnés d'une rubrique Amazon SNS. Les exemples d'instructions vous montrent comment configurer les autorisations CodeBuild permettant de communiquer avec Amazon SNS CloudWatch, comment créer et identifier des CodeBuild sujets dans Amazon SNS, comment inscrire des destinataires à ces sujets et comment configurer des règles dans. CloudWatch

Exemple Amazon ECR pour CodeBuild

Cet exemple utilise une image Docker dans un référentiel d'images Amazon Elastic Container Registry (Amazon ECR) pour créer un exemple de projet Go.

Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour AWS CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 AWS KMS, CloudWatch Logs et Amazon ECR. Pour plus d'informations, consultez [CodeBuild les rubriques Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) [Amazon et CloudWatch Tarification](#) [Amazon Elastic Container Registry](#).

Rubriques

- [Exécutez l'exemple Amazon ECR](#)

Exécutez l'exemple Amazon ECR

Suivez les instructions suivantes pour exécuter l'exemple Amazon ECR pour CodeBuild.

Pour exécuter cet exemple

1. Pour créer et transférer l'image Docker vers votre référentiel d'images dans Amazon ECR, suivez les étapes décrites dans la [Exécutez l'exemple « Publier une image Docker sur Amazon ECR »](#) section du. [Exemple « Publier une image Docker sur Amazon ECR »](#)

2. Créer un projet Go :

- a. Créez les fichiers comme décrit dans les [Fichiers d'un projet Go](#) sections [Structure d'un projet Go](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

- b. Créez un projet de build, exécutez le build et consultez les informations de build associées.

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "sample-go-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

```
}
```

- c. Pour obtenir le résultat de l'artefact de sortie de la génération, ouvrez votre compartiment de sortie S3.
 - d. Téléchargez le fichier *GoOutputArtifact.zip* sur votre ordinateur ou instance local, puis extrayez le contenu du fichier . Dans le contenu extrait, obtenez le fichier `hello`.
3. Si l'une des conditions suivantes est vraie, vous devez ajouter des autorisations à votre référentiel d'images dans Amazon ECR afin de AWS CodeBuild pouvoir intégrer son image Docker dans l'environnement de génération.
- Votre projet utilise des CodeBuild informations d'identification pour extraire les images Amazon ECR. Ceci est indiqué par la valeur `CODEBUILD` dans l'attribut `imagePullCredentialsType` de votre `ProjectEnvironment`.
 - Votre projet utilise une image Amazon ECR multi-comptes. Dans ce cas, votre projet doit utiliser son rôle de service pour extraire les images Amazon ECR. Pour activer ce comportement, définissez l'attribut `imagePullCredentialsType` de votre `ProjectEnvironment` sur `SERVICE_ROLE`.
1. Ouvrez la console Amazon ECR à <https://console.aws.amazon.com/ecr/> l'adresse.
 2. Dans la liste des noms de référentiel, choisissez le nom du référentiel que vous avez créé ou sélectionné.
 3. Dans le volet de navigation, choisissez Permissions (Autorisations), Edit (Modifier), puis Add statement (Ajouter une instruction).
 4. Pour Statement name, saisissez un identifiant (par exemple, **CodeBuildAccess**).
 5. Pour Effect (Effet), conservez la sélection Allow (Autoriser). Cela indique que vous souhaitez autoriser l'accès à un autre compte AWS .
 6. Pour Principal, effectuez l'une des opérations suivantes:
 - Si votre projet utilise des CodeBuild informations d'identification pour extraire une image Amazon ECR, saisissez **codebuild.amazonaws.com** Service principal.
 - Si votre projet utilise une image Amazon ECR multi-comptes IDs, saisissez pour AWS compte IDs les AWS comptes auxquels vous souhaitez donner accès.
 7. Ignorer la liste Toutes les entités IAM.
 8. Pour Action, sélectionnez les actions à extraction uniquement : `ecr :GetDownloadUrlForLayer`, `ecr : et ecr :BatchGetImage`. `BatchCheckLayerAvailability`

9. Pour Conditions, ajoutez ce qui suit :

```
{
  "StringEquals":{
    "aws:SourceAccount":"<AWS-account-ID>",
    "aws:SourceArn":"arn:aws:codebuild:<region>:<AWS-account-ID>:project/<project-name>"
  }
}
```

10. Choisissez Enregistrer.

Cette stratégie est affichée dans Autorisations. Le mandataire correspond à ce que vous avez saisi pour Principal (Mandataire) à l'étape 3 de cette procédure :

- Si votre projet utilise des CodeBuild informations d'identification pour extraire une image Amazon ECR, cela "codebuild.amazonaws.com" apparaît sous Principaux du service.
- Si votre projet utilise une image Amazon ECR multi-comptes, l'identifiant du AWS compte auquel vous souhaitez donner accès apparaît sous AWS Compte. IDs

L'exemple de politique suivant utilise à la fois des CodeBuild informations d'identification et une image Amazon ECR entre comptes.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:us-east-1:111122223333:project/MyProject",
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Sid": "CodeBuildAccessCrossAccount",
    "Effect": "Allow",
    "Action": [
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability"
    ],
    "Resource": "*"
  }
]
}

```

- Si vos projets utilisent des CodeBuild informations d'identification et que vous souhaitez qu'ils aient un accès ouvert au référentiel Amazon ECR, vous pouvez omettre les Condition clés et ajouter l'exemple de politique suivant. CodeBuild

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ],
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:codecommit:us-east-2:111122223333:MySharedDemoRepo"
      ]
    }
  ]
}

```

```

    ],
    "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
    ]
  }
}

```

4. Créez un projet de build, exécutez le build et consultez les informations de build.

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```

{
  "name": "amazon-ecr-sample-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "account-ID.dkr.ecr.region-ID.amazonaws.com/your-Amazon-ECR-repo-name:tag",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

5. Pour obtenir le résultat de l'artefact de sortie de la génération, ouvrez votre compartiment de sortie S3.

6. Téléchargez le fichier *GoOutputArtifact.zip* sur votre ordinateur ou instance local, puis extrayez le contenu du fichier *GoOutputArtifact.zip*. Dans le contenu extrait, obtenez le fichier `hello`.

Structure d'un projet Go

Cet exemple suppose que cette structure de répertoire est utilisée.

```
(root directory name)
### buildspec.yml
### hello.go
```

Fichiers d'un projet Go

Cet exemple utilise ces fichiers.

`buildspec.yml` (dans *(root directory name)*)

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
  build:
    commands:
      - echo Build started on `date`
      - echo Compiling the Go code
      - go build hello.go
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - hello
```

`hello.go` (dans *(root directory name)*)

```
package main
import "fmt"
```

```
func main() {  
    fmt.Println("hello world")  
    fmt.Println("1+1 =", 1+1)  
    fmt.Println("7.0/3.0 =", 7.0/3.0)  
    fmt.Println(true && false)  
    fmt.Println(true || false)  
    fmt.Println(!true)  
}
```

Exemple d'Amazon Elastic File System pour AWS CodeBuild

Vous souhaitez peut-être créer vos AWS CodeBuild versions sur Amazon Elastic File System, un service de fichiers partagé évolutif pour les instances Amazon EC2. La capacité de stockage d'Amazon EFS étant élastique, elle augmente ou diminue au fur et à mesure que des fichiers sont ajoutés ou supprimés. Il propose une interface de services Web simple, qui vous permet de créer et de configurer des systèmes de fichiers. Elle gère également l'ensemble de l'infrastructure de stockage de fichiers à votre place, de sorte que vous n'ayez pas besoin de vous préoccuper du déploiement, de l'application de correctifs ou de la maintenance des configurations de systèmes de fichiers. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon Elastic File System ?](#) dans le guide de l'utilisateur d'Amazon Elastic File System.

Cet exemple explique comment configurer un CodeBuild projet afin qu'il soit monté puis crée une application Java sur un système de fichiers Amazon EFS. Avant de commencer, vous devez disposer d'une application Java prête à être créée et téléchargée dans un compartiment d'entrée S3 AWS CodeCommit GitHub, un serveur GitHub d'entreprise ou un référentiel Bitbucket.

Les données en transit pour votre système de fichiers sont chiffrées. Pour chiffrer les données en transit à l'aide d'une autre image, veuillez consulter [Chiffrement des données en transit](#).

Rubriques

- [Utilisation AWS CodeBuild avec Amazon Elastic File System](#)
- [Résoudre les problèmes liés à l'intégration Amazon EFS](#)

Utilisation AWS CodeBuild avec Amazon Elastic File System

L'exemple couvre les quatre étapes de haut niveau requises pour utiliser Amazon EFS avec AWS CodeBuild. Il s'agit des options suivantes :

1. Créez un cloud privé virtuel (VPC) dans votre AWS compte.
2. Créez un système de fichiers qui utilise ce VPC.
3. Créez et créez un CodeBuild projet qui utilise le VPC. Le CodeBuild projet utilise les éléments suivants pour identifier le système de fichiers :
 - Un identificateur de système de fichiers unique. Vous choisissez cet identificateur lorsque vous spécifiez le système de fichiers dans votre projet de génération.
 - ID du système de fichiers. L'ID s'affiche lorsque vous consultez votre système de fichiers dans la console Amazon EFS.
 - Un point de montage. Il s'agit d'un répertoire dans votre conteneur Docker qui monte le système de fichiers.
 - Des options de montage. Elles incluent des détails sur la façon de monter le système de fichiers.
4. Passez en revue le projet de construction pour vous assurer que les fichiers de projet et les variables appropriés ont été générés.

Note

Un système de fichiers créé dans Amazon EFS est uniquement pris en charge sur les plateformes Linux.

Rubriques

- [Étape 1 : créer un VPC à l'aide de CloudFormation](#)
- [Étape 2 : créer un système de fichiers Amazon Elastic File System avec votre VPC](#)
- [Étape 3 : créer un CodeBuild projet à utiliser avec Amazon EFS](#)
- [Étape 4 : Passez en revue le projet de construction](#)

Étape 1 : créer un VPC à l'aide de CloudFormation

Créez votre VPC à l'aide d'un CloudFormation modèle.

1. Suivez les instructions d'utilisation [CloudFormation Modèle VPC](#) pour CloudFormation créer un VPC.

 Note

Le VPC créé par ce CloudFormation modèle possède deux sous-réseaux privés et deux sous-réseaux publics. Vous ne devez utiliser des sous-réseaux privés que lorsque vous montez le système AWS CodeBuild de fichiers que vous avez créé dans Amazon EFS. Si vous utilisez l'un des sous-réseaux publics, la génération échoue.

2. Connectez-vous à la console Amazon VPC AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/vpc/>
3. Choisissez le VPC avec lequel vous avez créé. CloudFormation
4. Dans l'onglet Description, notez le nom de votre VPC et son ID. Les deux seront requis lorsque vous créerez votre projet AWS CodeBuild plus tard dans cet exemple.

Étape 2 : créer un système de fichiers Amazon Elastic File System avec votre VPC

Créez un système de fichiers Amazon EFS simple pour cet exemple à l'aide du VPC que vous avez créé précédemment.

1. Connectez-vous à la console Amazon EFS AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/efs/>.
2. Choisissez Create file system (Créer un système de fichiers).
3. Dans VPC, choisissez le nom de VPC que vous avez noté précédemment dans cet exemple.
4. Conservez les Zones de disponibilité associées à vos sous-réseaux sélectionnés.
5. Choisissez Étape suivante.
6. Dans Ajouter des balises, pour la clé Nom par défaut, dans Valeur, entrez le nom de votre système de fichiers Amazon EFS.
7. Conservez Bursting (Transmission en rafales) et General Purpose (Usage général) sélectionnés en tant que modes de performance et de débit par défaut, puis choisissez Étape suivante.
8. Pour Configurer client access (Configurer l'accès client), choisissez Next Step (Étape suivante).
9. Sélectionnez Créer un système de fichiers.
10. (Facultatif) Nous vous recommandons d'ajouter une politique à votre système de fichiers Amazon EFS qui applique le chiffrement des données en transit. Dans la console Amazon EFS, choisissez la politique du système de fichiers, choisissez Modifier, cochez la case intitulée Appliquer le chiffrement en transit pour tous les clients, puis sélectionnez Enregistrer.

Étape 3 : créer un CodeBuild projet à utiliser avec Amazon EFS

Créez un AWS CodeBuild projet qui utilise le VPC que vous avez créé précédemment dans cet exemple. Lorsque le build est exécuté, il monte le système de fichiers Amazon EFS créé précédemment. Ensuite, elle stocke le fichier .jar créé par votre application Java dans le répertoire de point de montage de votre système de fichiers.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Choisissez Build projects (Projets de génération) dans le volet de navigation, puis choisissez Create a build project (Créer un projet de build).
3. Dans Project name (Nom de projet), saisissez un nom pour votre projet.
4. Dans Source provider (Fournisseur de source), choisissez le référentiel qui contient l'application Java que vous voulez générer.
5. Entrez les informations, telles que l'URL du référentiel, CodeBuild qui permettent de localiser votre application. Les options sont différentes pour chaque fournisseur de source. Pour de plus amples informations, veuillez consulter [Choose source provider](#).
6. Dans Image de l'environnement, choisissez Image gérée.
7. Dans Operating system (Système d'exploitation), choisissez Amazon Linux 2.
8. Dans Runtime(s) (Exécution(s)), sélectionnez Standard.
9. Dans Image, choisissez aws/codebuild/amazonlinux-x86_64-standard:4.0.
10. Dans Environment type (Type d'environnement), choisissez Linux.
11. Dans Service role (Rôle de service), choisissez New service role (Nouveau rôle de service). Dans Nom du rôle, entrez le nom du rôle CodeBuild créé pour vous.
12. Développez Additional configuration (Configuration supplémentaire).
13. Sélectionnez Enable this flag if you want to build Docker images or want your builds to get elevated privileges (Activer cet indicateur si vous souhaitez créer des images Docker ou que vos builds reçoivent des privilèges élevés).

Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux](#) Capabilities sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

14. Choisissez l'ID de VPC dans VPC.
15. Dans Subnets (Sous-réseaux), choisissez un ou plusieurs sous-réseaux privés associés à votre VPC. Vous devez utiliser des sous-réseaux privés dans une version qui monte un système de fichiers Amazon EFS. La génération échoue si vous utilisez un sous-réseau public.
16. Pour Security Groups (Groupes de sécurité), choisissez le groupe de sécurité par défaut.
17. Dans File systems (Systèmes de fichiers), entrez les informations suivantes :
 - Pour Identifier (Identificateur), entrez un identificateur de système de fichiers unique. Il doit comporter moins de 129 caractères et ne contenir que des caractères alphanumériques et des traits de soulignement. CodeBuild utilise cet identificateur pour créer une variable d'environnement qui identifie le système de fichiers élastique. Le format de variable d'environnement est `CODEBUILD_<file_system_identifieur>` en lettres majuscules. Par exemple, si vous entrez `my_efs`, la variable d'environnement est `CODEBUILD_MY_EFS`.
 - Pour ID, choisissez l'ID du système de fichiers.
 - (Facultatif) Entrez un répertoire dans le système de fichiers. CodeBuild monte ce répertoire. Si vous laissez Directory path (Chemin du répertoire) vide, CodeBuild monte le système de fichiers complet. Le chemin est relatif à la racine du système de fichiers.
 - Pour Point de montage, entrez le chemin absolu du répertoire dans votre conteneur de compilation où le système de fichiers est monté. Si ce répertoire n'existe pas, CodeBuild créez-le lors de la compilation.
 - (Facultatif) Entrez des options de montage. Si vous laissez le champ Options de montage vide, CodeBuild utilise ses options de montage par défaut :

```
nfsvers=4.1
rsize=1048576
wsize=1048576
hard
timeo=600
retrans=2
```

Pour plus d'informations, consultez les [options de montage NFS recommandées](#) dans le guide de l'utilisateur d'Amazon Elastic File System.

18. Pour Build spécification (Spécification de génération), choisissez Insert build commands (Insérer les commandes de génération), puis choisissez Switch to editor (Basculer vers l'éditeur).

19. Entrez les commandes de spécification de construction suivantes dans l'éditeur. Remplacez `<file_system_identifieur>` par l'identificateur que vous avez entré à l'étape 17. Utilisez des lettres majuscules (par exemple, CODEBUILD_MY_EFS).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
  build:
    commands:
      - mvn compile -Dgpg.skip=true -Dmaven.repo.local=
$CODEBUILD_<file_system_identifieur>
```

20. Utilisez les valeurs par défaut pour les autres paramètres, puis choisissez Create build project (Créer un projet de build). Lorsque votre génération est terminée, la page de la console correspondant à votre projet est affichée.
21. Choisissez Démarrer la génération.

Étape 4 : Passez en revue le projet de construction

Une fois votre AWS CodeBuild projet créé :

- Vous disposez d'un fichier .jar créé par votre application Java qui est intégré à votre système de fichiers Amazon EFS dans le répertoire de votre point de montage.
- Une variable d'environnement qui identifie votre système de fichiers est créée à l'aide de l'identificateur de système de fichiers que vous avez entré lors de la création du projet.

Pour plus d'informations, consultez la section [Montage de systèmes de fichiers](#) dans le manuel Amazon Elastic File System User Guide.

Résoudre les problèmes liés à l'intégration Amazon EFS

Les erreurs suivantes sont susceptibles de se produire lors de la configuration d'Amazon EFS avec CodeBuild.

Rubriques

- [CLIENT_ERROR : échec du montage de « 127.0.0.1 :/». Autorisation refusée](#)

- [CLIENT_ERROR : échec du montage de '127.0.0.1 :/'. Connexion réinitialisée par un pair](#)
- [VPC_CLIENT_ERROR : erreur EC2 inattendue : UnauthorizedOperation](#)

CLIENT_ERROR : échec du montage de « 127.0.0.1 :/ ». Autorisation refusée

L'autorisation IAM n'est pas prise en charge pour le montage d'Amazon EFS avec CodeBuild. Si vous utilisez une politique de système de fichiers Amazon EFS personnalisée, vous devez accorder un accès en lecture et en écriture à tous les principaux IAM. Par exemple :

```
"Principal": {  
  "AWS": "*"   
}
```

CLIENT_ERROR : échec du montage de '127.0.0.1 :/'. Connexion réinitialisée par un pair

Cette erreur peut avoir deux causes :

- Le sous-réseau CodeBuild VPC se trouve dans une zone de disponibilité différente de celle de la cible de montage Amazon EFS. Vous pouvez résoudre ce problème en ajoutant un sous-réseau VPC dans la même zone de disponibilité que la cible de montage Amazon EFS.
- Le groupe de sécurité n'est pas autorisé à communiquer avec Amazon EFS. Vous pouvez résoudre ce problème en ajoutant une règle entrante pour autoriser tout le trafic provenant du VPC (ajoutez le bloc CIDR principal pour votre VPC) ou du groupe de sécurité lui-même.

VPC_CLIENT_ERROR : erreur EC2 inattendue : UnauthorizedOperation

Cette erreur se produit lorsque tous les sous-réseaux de votre configuration VPC pour CodeBuild le projet sont des sous-réseaux publics. Vous devez disposer d'au moins un sous-réseau privé dans le VPC pour garantir la connectivité réseau.

AWS CodePipeline échantillons pour CodeBuild

Cette section décrit des exemples d'intégrations entre CodePipeline et CodeBuild

Exemple	Description
Exemples d' CodePipeline/CodeBuild intégrations et de versions par lots	Ces exemples montrent comment AWS CodePipeline créer un projet de génération utilisant des builds par lots.
Exemple d' CodePipeline/CodeBuild intégration avec plusieurs sources d'entrée et artefacts de sortie	Cet exemple montre comment AWS CodePipeline créer un projet de génération qui utilise plusieurs sources d'entrée pour créer plusieurs artefacts de sortie.

Exemples d' CodePipeline/CodeBuild intégrations et de versions par lots

AWS CodeBuild prend en charge les builds par lots. Les exemples suivants montrent comment AWS CodePipeline créer un projet de génération utilisant des builds par lots.

Vous pouvez utiliser un fichier au format JSON qui définit la structure de votre pipeline, puis l'utiliser avec le AWS CLI pour créer le pipeline. Pour plus d'informations, consultez la section [Référence sur la structure du AWS CodePipeline pipeline](#) dans le guide de AWS CodePipeline l'utilisateur.

Construction par lots avec des artefacts individuels

Utilisez le fichier JSON suivant comme exemple de structure de pipeline qui crée une construction par lots avec des artefacts distincts. Pour activer les intégrations par lots CodePipeline, définissez le BatchEnabled paramètre de l'configurationobjet surtrue.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
```

```
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "source1"
        }
      ],
      "configuration": {
        "S3Bucket": "<my-input-bucket-name>",
        "S3ObjectKey": "my-source-code-file-name.zip"
      },
      "runOrder": 1
    },
    {
      "inputArtifacts": [],
      "name": "Source2",
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "version": "1",
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "source2"
        }
      ],
      "configuration": {
        "S3Bucket": "<my-other-input-bucket-name>",
        "S3ObjectKey": "my-other-source-code-file-name.zip"
      },
      "runOrder": 1
    }
  ]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        }
      ],
      {
```



```
        "name": "source2"
      },
    ],
    "name": "Build",
    "actionTypeId": {
      "category": "Build",
      "owner": "AWS",
      "version": "1",
      "provider": "CodeBuild"
    },
    "outputArtifacts": [
      {
        "name": "build1"
      },
      {
        "name": "build1_artifact1"
      },
      {
        "name": "build1_artifact2"
      },
      {
        "name": "build2_artifact1"
      },
      {
        "name": "build2_artifact2"
      }
    ],
    "configuration": {
      "ProjectName": "my-build-project-name",
      "PrimarySource": "source1",
      "BatchEnabled": "true"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
```

```
}
```

Voici un exemple de fichier CodeBuild buildspec qui fonctionnera avec cette configuration de pipeline.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
  secondary-artifacts:
    artifact1:
      files:
        - output_file
    artifact2:
      files:
        - output_file
```

Les noms des artefacts de sortie spécifiés dans le fichier JSON du pipeline doivent correspondre à l'identifiant des builds et des artefacts définis dans votre fichier buildspec. La syntaxe est *buildIdentifier* pour les artefacts principaux et *buildIdentifier_artifactIdentifier* pour les artefacts secondaires.

Par exemple, pour le nom de l'artefact en sortie build1, CodeBuild téléchargera l'artefact principal de build1 à l'emplacement de. build1 Pour le nom de sortie build1_artifact1, CodeBuild téléchargera l'artefact secondaire artifact1 de build1 à l'emplacement de build1_artifact1, et ainsi de suite. Si un seul emplacement de sortie est spécifié, le nom doit être *buildIdentifier* uniquement.

Une fois que vous avez créé le fichier JSON, vous pouvez créer votre pipeline. Utilisez le AWS CLI pour exécuter la commande `create-pipeline` et transmettre le fichier au `--cli-input-json` paramètre. Pour plus d'informations, consultez la section [Créer un pipeline \(CLI\)](#) dans le guide de AWS CodePipeline l'utilisateur.

Construction par lots avec des artefacts combinés

Utilisez le fichier JSON suivant comme exemple de structure de pipeline qui crée une construction par lots avec des artefacts combinés. Pour activer les intégrations par lots CodePipeline, définissez le `BatchEnabled` paramètre de l'configurationobjet sur `true`. Pour combiner les artefacts de construction au même endroit, définissez le `CombineArtifacts` paramètre de l'configurationobjet sur `true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "<my-input-bucket-name>",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          },
          {
            "inputArtifacts": [],
```

```
    "name": "Source2",
    "actionTypeId": {
      "category": "Source",
      "owner": "AWS",
      "version": "1",
      "provider": "S3"
    },
    "outputArtifacts": [
      {
        "name": "source2"
      }
    ],
    "configuration": {
      "S3Bucket": "<my-other-input-bucket-name>",
      "S3ObjectKey": "my-other-source-code-file-name.zip"
    },
    "runOrder": 1
  }
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "output1 "
        }
      ]
    },
  ],
}
```

```
    "configuration": {
      "ProjectName": "my-build-project-name",
      "PrimarySource": "source1",
      "BatchEnabled": "true",
      "CombineArtifacts": "true"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

Voici un exemple de fichier CodeBuild buildspec qui fonctionnera avec cette configuration de pipeline.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
```

Si les artefacts combinés sont activés pour la génération par lots, une seule sortie est autorisée. CodeBuild combinera les principaux artefacts de toutes les versions dans un seul fichier ZIP.

Une fois que vous avez créé le fichier JSON, vous pouvez créer votre pipeline. Utilisez le AWS CLI pour exécuter la commande `create-pipeline` et transmettre le fichier au `--cli-input-json` paramètre. Pour plus d'informations, consultez la section [Créer un pipeline \(CLI\)](#) dans le guide de AWS CodePipeline l'utilisateur.

Exemple d' CodePipeline/CodeBuild intégration avec plusieurs sources d'entrée et artefacts de sortie

Un AWS CodeBuild projet peut utiliser plusieurs sources d'entrée. Il peut également créer plusieurs artefacts de sortie. Cet exemple montre comment AWS CodePipeline créer un projet de génération qui utilise plusieurs sources d'entrée pour créer plusieurs artefacts de sortie. Pour de plus amples informations, veuillez consulter [Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#).

Vous pouvez utiliser un fichier au format JSON qui définit la structure de votre pipeline, puis l'utiliser avec le AWS CLI pour créer le pipeline. Utilisez le fichier JSON suivant comme exemple de structure de pipeline qui crée une génération avec plusieurs sources d'entrée et plusieurs artefacts de sortie. Ultérieurement dans cet exemple, vous apprendrez comment ce fichier spécifie les différentes entrées et sorties. Pour plus d'informations, consultez [la référence sur la structure du CodePipeline pipeline](#) dans le guide de AWS CodePipeline l'utilisateur.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "configuration": {
      "S3Bucket": "my-input-bucket-name",
      "S3ObjectKey": "my-source-code-file-name.zip"
    },
    "runOrder": 1
  },
  {
    "inputArtifacts": [],
    "name": "Source2",
    "actionTypeId": {
      "category": "Source",
      "owner": "AWS",
      "version": "1",
      "provider": "S3"
    },
    "outputArtifacts": [
      {
        "name": "source2"
      }
    ],
    "configuration": {
      "S3Bucket": "my-other-input-bucket-name",
      "S3ObjectKey": "my-other-source-code-file-name.zip"
    },
    "runOrder": 1
  }
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",

```

```
    "owner": "AWS",
    "version": "1",
    "provider": "AWS CodeBuild"
  },
  "outputArtifacts": [
    {
      "name": "artifact1"
    },
    {
      "name": "artifact2"
    }
  ],
  "configuration": {
    "ProjectName": "my-build-project-name",
    "PrimarySource": "source1"
  },
  "runOrder": 1
}
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "AWS-CodePipeline-internal-bucket-name"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

Dans ce fichier JSON :

- L'une de vos sources d'entrée doit être désignée comme `PrimarySource`. Cette source est le répertoire dans lequel CodeBuild recherche et exécute votre fichier `buildspec`. Le mot clé `PrimarySource` est utilisé pour spécifier la source principale dans la `configuration` section du CodeBuild stage du fichier JSON.
- Chaque source d'entrée est installée dans son propre répertoire. Ce répertoire est stocké dans la variable d'environnement intégrée `$CODEBUILD_SRC_DIR` pour la source principale et `$CODEBUILD_SRC_DIR_YourInputArtifactName` pour toutes les autres sources. Pour le pipeline de cet exemple, les deux répertoires des sources d'entrée sont `$CODEBUILD_SRC_DIR` et

`$CODEBUILD_SRC_DIR_source2`. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

- Les noms des artefacts de sortie spécifiés dans le fichier JSON du pipeline doivent correspondre aux noms des artefacts secondaires définis dans votre fichier buildspec. Ce pipeline utilise le fichier buildspec suivant. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

```
version: 0.2

phases:
  build:
    commands:
      - touch source1_file
      - cd $CODEBUILD_SRC_DIR_source2
      - touch source2_file

artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR
      files:
        - source1_file
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - source2_file
```

Une fois que vous avez créé le fichier JSON, vous pouvez créer votre pipeline. Utilisez le AWS CLI pour exécuter la commande `create-pipeline` et transmettre le fichier au `--cli-input-json` paramètre. Pour plus d'informations, consultez la section [Créer un pipeline \(CLI\)](#) dans le guide de AWS CodePipeline l'utilisateur.

AWS Config échantillon avec CodeBuild

AWS Config fournit un inventaire de vos AWS ressources et un historique des modifications de configuration apportées à ces ressources. AWS Config prend désormais en charge en AWS CodeBuild tant que AWS ressource, ce qui signifie que le service peut suivre vos CodeBuild projets.

Pour plus d'informations AWS Config, voir [Qu'est-ce que c'est AWS Config ?](#) dans le Guide AWS Config du développeur.

Vous pouvez consulter les informations suivantes concernant les CodeBuild ressources sur la page Inventaire des ressources de la AWS Config console :

- Chronologie de vos modifications CodeBuild de configuration.
- Détails de configuration pour chaque CodeBuild projet.
- Relations avec d'autres AWS ressources.
- Liste des modifications apportées à vos CodeBuild projets.

Rubriques

- [À utiliser CodeBuild avec AWS Config](#)
- [Étape 3 : Afficher AWS CodeBuild les données dans la AWS Config console](#)

À utiliser CodeBuild avec AWS Config

Les procédures décrites dans cette rubrique vous montrent comment configurer AWS Config et rechercher CodeBuild des projets.

Rubriques

- [Conditions préalables](#)
- [Étape 1 : Configuration AWS Config](#)
- [Étape 2 : Rechercher AWS CodeBuild des projets](#)

Conditions préalables

Créez votre AWS CodeBuild projet. Pour obtenir des instructions, veuillez consulter [Création d'un projet de génération](#).

Étape 1 : Configuration AWS Config

- [Configuration de AWS Config \(console\)](#)
- [Configuration de AWS Config \(AWS CLI\)](#)

 Note

Une fois la configuration terminée, l'affichage des AWS CodeBuild projets dans la AWS Config console peut prendre jusqu'à 10 minutes.

Étape 2 : Rechercher AWS CodeBuild des projets

1. Connectez-vous à la console de AWS gestion et ouvrez la AWS Config console à l'adresse <https://console.aws.amazon.com/config>.
2. Sur la page Inventaire des ressources, sélectionnez AWS CodeBuild Projet sous Type de ressource. Faites défiler la page vers le bas et cochez la case CodeBuilddu projet.
3. Choisissez Recherche.
4. Une fois la liste des CodeBuild projets ajoutée, cliquez sur le lien du nom du CodeBuild projet dans la colonne Config timeline.

Étape 3 : Afficher AWS CodeBuild les données dans la AWS Config console

Lorsque vous recherchez des ressources sur la page Inventaire des ressources, vous pouvez choisir la AWS Config chronologie pour afficher les détails de votre CodeBuild projet. La page des détails d'une ressource fournit des informations sur la configuration, les relations et le nombre de modifications apportées à cette ressource.

Les blocs situés en haut de la page constituent la chronologie. La chronologie affiche la date et l'heure de l'enregistrement.

Pour plus d'informations, consultez la section [Affichage des détails de configuration dans la AWS Config console](#) dans le Guide du AWS Config développeur.

Créer un exemple de notifications pour CodeBuild

Amazon CloudWatch Events dispose d'un support intégré pour AWS CodeBuild. CloudWatch Les événements sont un flux d'événements système décrivant les modifications apportées à vos AWS ressources. Avec CloudWatch Events, vous rédigez des règles déclaratives pour associer les événements intéressants aux actions automatisées à entreprendre. Cet exemple utilise Amazon CloudWatch Events et Amazon Simple Notification Service (Amazon SNS) pour envoyer des notifications de build aux abonnés chaque fois que les builds réussissent, échouent, passent d'une phase de construction à une autre, ou toute combinaison de ces événements.

⚠ Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon CloudWatch et Amazon SNS. Pour plus d'informations, consultez les [CodeBuild sections Tarification](#), [CloudWatchTarification Amazon](#) et Tarification [Amazon SNS](#).

Rubriques

- [Exécutez l'exemple de notifications de build](#)
- [Référence du format d'entrée des notifications de génération](#)

Exécutez l'exemple de notifications de build

Utilisez la procédure suivante pour exécuter l'exemple de notifications de génération.

Pour exécuter cet exemple

1. Si vous avez déjà configuré une rubrique et que vous êtes abonné sur Amazon SNS et que vous souhaitez utiliser pour cet exemple, passez à l'étape 4. Sinon, si vous utilisez un utilisateur IAM au lieu d'un compte AWS root ou un utilisateur administrateur pour travailler avec Amazon SNS, ajoutez la déclaration suivante (**### BEGIN ADDING STATEMENT HERE ###**entre **### END ADDING STATEMENT HERE ###** et) à l'utilisateur (ou au groupe IAM auquel l'utilisateur est associé). L'utilisation d'un compte AWS root n'est pas recommandée. Cette déclaration permet de consulter, de créer, de s'abonner et de tester l'envoi de notifications à des rubriques sur Amazon SNS. Des ellipses (. . .) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter l'instruction. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie existante.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
```

```
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
    ],
    "Resource": "*"
}
]
```

Note

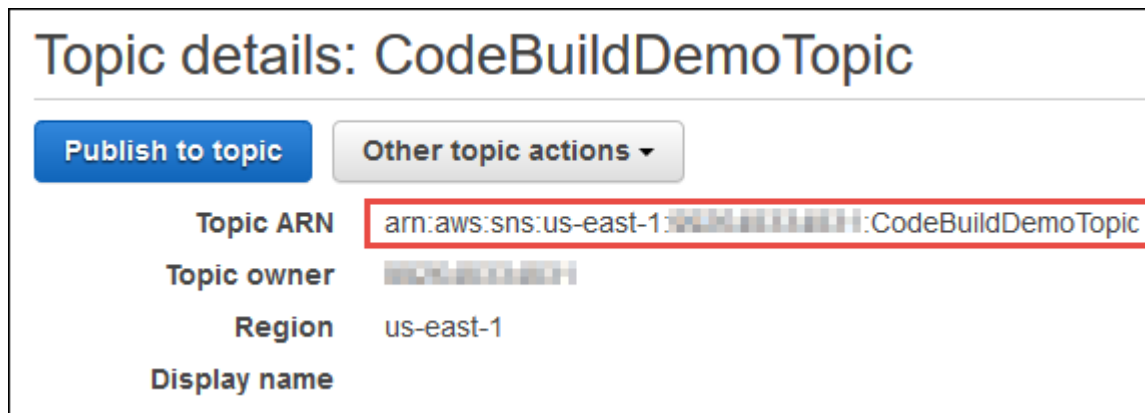
L'entité IAM qui modifie cette politique doit être autorisée dans IAM à modifier les politiques.

Pour plus d'informations, consultez la section [Modification des politiques gérées par le client](#) ou la section « Pour modifier ou supprimer une politique intégrée pour un groupe, un utilisateur ou un rôle » dans [Utilisation des politiques intégrées \(console\)](#) du guide de l'utilisateur IAM.

2. Créez ou identifiez un sujet dans Amazon SNS. AWS CodeBuild utilise CloudWatch Events pour envoyer des notifications de build à cette rubrique via Amazon SNS.

Pour créer une rubrique:

1. [Ouvrez la console Amazon SNS à l'adresse /sns. https://console.aws.amazon.com](https://console.aws.amazon.com/sns)
2. Choisissez Créer une rubrique.
3. Dans Create new topic (Créer une rubrique), pour Topic name (Nom de rubrique), saisissez un nom de rubrique (par exemple, **CodeBuildDemoTopic**). (Si vous choisissez un autre nom, remplacez-le dans l'ensemble de cet exemple.)
4. Choisissez Créer une rubrique.
5. Sur la CodeBuildDemoTopic page Détails du sujet :, copiez la valeur ARN du sujet. Vous avez besoin de cette valeur pour l'étape suivante.



Topic details: CodeBuildDemoTopic

Publish to topic **Other topic actions ▾**

Topic ARN `arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic`

Topic owner `123456789012`

Region `us-east-1`

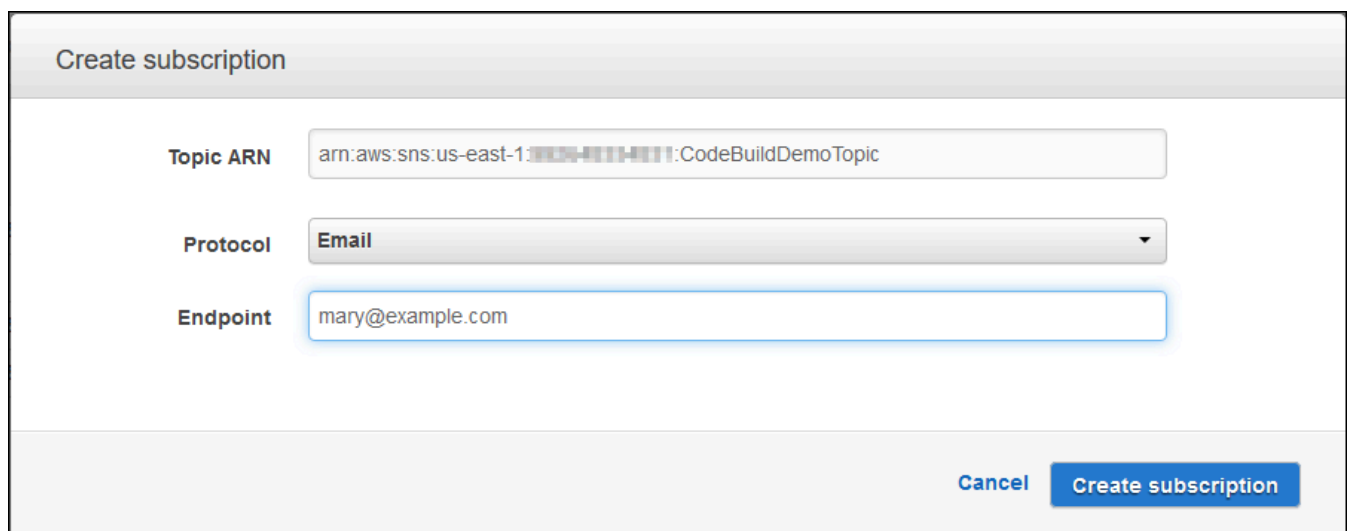
Display name

Pour plus d'informations, consultez la section [Créer une rubrique](#) dans le manuel Amazon SNS Developer Guide.

3. Abonnez un ou plusieurs destinataires à la rubrique pour recevoir des notifications par e-mail.

Pour abonner un destinataire à une rubrique :

1. La console Amazon SNS étant ouverte depuis l'étape précédente, dans le volet de navigation, choisissez Subscriptions, puis Create subscription.
2. Dans la boîte de dialogue Create subscription (Créer un abonnement, pour Topic ARN (ARN de la rubrique), collez l'ARN de la rubrique que vous avez copié à l'étape précédente.
3. Pour Protocole, choisissez E-mail.
4. Indiquez l'adresse e-mail complète du destinataire dans Endpoint (Point de terminaison).



Create subscription

Topic ARN `arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic`

Protocol `Email`

Endpoint `mary@example.com`

Cancel **Create subscription**

5. Choisissez Create Subscription (Créer un abonnement).
6. Amazon SNS envoie un e-mail de confirmation d'abonnement au destinataire. Pour commencer à recevoir des notifications par e-mail, le destinataire doit choisir le lien Confirmer l'abonnement dans l'e-mail de confirmation d'abonnement. Une fois que le destinataire a cliqué sur le lien, s'il est correctement inscrit, Amazon SNS affiche un message de confirmation dans le navigateur Web du destinataire.

Pour plus d'informations, consultez la section [S'abonner à une rubrique](#) du manuel Amazon SNS Developer Guide.

4. Si vous utilisez un utilisateur plutôt qu'un compte AWS root ou un utilisateur administrateur pour travailler avec CloudWatch Events, ajoutez l'instruction suivante (entre **### BEGIN ADDING STATEMENT HERE ###** et **### END ADDING STATEMENT HERE ###**) à l'utilisateur (ou au groupe IAM auquel l'utilisateur est associé). L'utilisation d'un compte AWS root n'est pas recommandée. Cette déclaration est utilisée pour permettre à l'utilisateur de travailler avec CloudWatch Events. Des ellipses (. . .) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter l'instruction. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie existante.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/Service*"
    }
  ]
}
```

 Note

L'entité IAM qui modifie cette politique doit être autorisée dans IAM à modifier les politiques.

Pour plus d'informations, consultez la section [Modification des politiques gérées par le client](#) ou la section « Pour modifier ou supprimer une politique intégrée pour un groupe, un utilisateur ou un rôle » dans [Utilisation des politiques intégrées \(console\)](#) du guide de l'utilisateur IAM.

5. Créez une règle dans CloudWatch Events. Pour ce faire, ouvrez la CloudWatch console, dans <https://console.aws.amazon.com/cloudwatch>.
6. Dans le volet de navigation, sous Events, choisissez Rules, puis Create rule.
7. Sur la page Step 1: Create rule page (Étape 1 : Créer une page de règle), les options Event Pattern (Modèle d'événement) et Build event pattern to match events by service (Créer un modèle d'événement pour correspondre aux événements en fonction du service) doivent toujours être activées.
8. Pour Service Name (Nom du service), choisissez CodeBuild. Pour Event Type (Type d'événement), vous devez toujours choisir All Events (Tous les événements).
9. Le code suivant doit être affiché dans l'aperçu du modèle d'événement :

```
{
  "source": [
    "aws.codebuild"
  ]
}
```

10. Choisissez Edit (Modifier) pour remplacer le code présent dans Event Pattern Preview (Aperçu du modèle d'événement) par l'un des deux modèles de règle suivants.

Ce premier modèle de règle déclenche un événement chaque fois qu'une génération commence ou se termine, pour les projets de génération spécifiés dans AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
```



```
"CodeBuild Build State Change"
],
"detail": {
  "build-status": [
    "IN_PROGRESS",
    "SUCCEEDED",
    "FAILED",
    "STOPPED"
  ],
  "project-name": [
    "my-demo-project-1",
    "my-demo-project-2"
  ]
}
}
```

Dans la règle précédente, effectuez les modifications de code suivantes en fonction de vos besoins.

- Pour déclencher un événement chaque fois qu'une génération commence ou se termine, laissez toutes les valeurs indiquées dans la zone `build-status` ou supprimez complètement la zone `build-status`.
- Pour déclencher un événement uniquement lorsqu'une génération est terminée, supprimez `IN_PROGRESS` dans la zone `build-status`.
- Pour déclencher un événement uniquement lorsqu'une génération commence, supprimez toutes les valeurs sauf `IN_PROGRESS` dans la zone `build-status`.
- Pour déclencher des événements inhérents à tous les projets de génération, supprimez complètement la zone `project-name`.
- Pour déclencher des événements uniquement pour les projets de génération individuels, spécifiez le nom de chaque projet de génération dans la zone `project-name`.

Ce second modèle de règle déclenche un événement chaque fois qu'une génération passe d'une phase à une autre pour les projets de génération spécifiés dans AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
```

```
"CodeBuild Build Phase Change"
],
"detail": {
  "completed-phase": [
    "SUBMITTED",
    "PROVISIONING",
    "DOWNLOAD_SOURCE",
    "INSTALL",
    "PRE_BUILD",
    "BUILD",
    "POST_BUILD",
    "UPLOAD_ARTIFACTS",
    "FINALIZING"
  ],
  "completed-phase-status": [
    "TIMED_OUT",
    "STOPPED",
    "FAILED",
    "SUCCEEDED",
    "FAULT",
    "CLIENT_ERROR"
  ],
  "project-name": [
    "my-demo-project-1",
    "my-demo-project-2"
  ]
}
}
```

Dans la règle précédente, effectuez les modifications de code suivantes en fonction de vos besoins.

- Pour déclencher un événement à chaque changement de phase de génération (ce qui peut générer l'envoi de jusqu'à 9 notifications par génération), laissez toutes les valeurs indiquées dans la zone `completed-phase` ou supprimez complètement le tableau `completed-phase`.
- Pour déclencher des événements uniquement lors des changements de phase de génération individuelle, supprimez le nom de chaque phase de génération pour laquelle vous ne souhaitez pas déclencher un événement dans la zone `completed-phase`.
- Pour déclencher un événement à chaque changement de statut de phase de génération, laissez toutes les valeurs indiquées dans la zone `completed-phase-status` ou supprimez complètement la zone `completed-phase-status`.

- Pour déclencher des événements uniquement lors des changements de statut de phase de génération individuelle, supprimez le nom de chaque statut de phase de génération pour lequel vous ne souhaitez pas déclencher un événement dans la zone `completed-phase-status`.
- Pour déclencher des événements pour tous les projets de génération, supprimez la zone `project-name`.
- Pour déclencher des événements pour les projets de génération individuels, spécifiez le nom de chaque projet de génération dans la zone `project-name`.

Pour plus d'informations sur les modèles d'événements, consultez la section [Modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations sur le filtrage à l'aide de modèles d'événements, consultez la section [Filtrage basé sur le contenu avec des modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

 Note

Si vous souhaitez déclencher des événements à la fois pour les changements d'état de génération et les changements de phase de génération, vous devez créer deux règles distinctes, une pour les changements d'état de génération et une autre pour les changements de phase de génération. Si vous essayez de combiner les deux règles en une seule, la règle combinée peut entraîner des résultats inattendus ou un arrêt de la procédure.

Une fois que vous avez terminé le remplacement du code, choisissez Save.

11. Pour Targets (Cibles), choisissez Add target (Ajouter une cible).
12. Dans la liste des cibles, sélectionnez Rubrique SNS.
13. Pour Topic, choisissez la rubrique que vous avez identifiée ou créée précédemment.
14. Développez Configurer l'entrée, puis choisissez Transformateur d'entrée.
15. Dans la zone Input Path (Chemin d'entrée), tapez l'un des chemins d'entrée ci-après.

Pour une règle dont l'élément `detail-type` a pour valeur `CodeBuild Build State Change`, tapez ce qui suit.

```
{"build-id":"$.detail.build-id","project-name":"$.detail.project-name","build-status":"$.detail.build-status"}
```

Pour une règle dont l'élément `detail-type` a pour valeur `CodeBuild Build Phase Change`, tapez ce qui suit.

```
{"build-id":"$.detail.build-id","project-name":"$.detail.project-name","completed-phase":"$.detail.completed-phase","completed-phase-status":"$.detail.completed-phase-status"}
```

Pour obtenir d'autres types d'informations, consultez [Référence du format d'entrée des notifications de génération](#).

16. Dans la zone Input Template (Modèle d'entrée), tapez l'un des modèles d'entrée ci-après.

Pour une règle dont l'élément `detail-type` a pour valeur `CodeBuild Build State Change`, tapez ce qui suit.

```
"Build '<build-id>' for build project '<project-name>' has reached the build status of '<build-status>'."
```

Pour une règle dont l'élément `detail-type` a pour valeur `CodeBuild Build Phase Change`, tapez ce qui suit.

```
"Build '<build-id>' for build project '<project-name>' has completed the build phase of '<completed-phase>' with a status of '<completed-phase-status>'."
```

17. Choisissez Configure details (Configurer les détails).
18. Sur la page Step 2: Configure rule details (Étape 2 : Configurer les détails de la règle), tapez un nom et une description. Pour State (État), la case Enabled (Activé) doit rester sélectionnée.
19. Choisissez Créer une règle.
20. Créez des projets de build, exécutez les builds et consultez les informations de build.
21. Confirmez CodeBuild que les notifications de build sont désormais correctement envoyées. Par exemple, vérifiez si les e-mails de notification de build sont à présent dans votre boîte de réception.

Pour modifier le comportement d'une règle, dans la CloudWatch console, choisissez la règle que vous souhaitez modifier, sélectionnez Actions, puis sélectionnez Modifier. Modifiez la règle, choisissez Configure details (Configurer les détails), puis choisissez Mettre à jour la règle (Update rule).

Pour arrêter d'utiliser une règle pour envoyer des notifications de build, dans la CloudWatch console, choisissez la règle que vous souhaitez arrêter d'utiliser, choisissez Actions, puis choisissez Désactiver.

Pour supprimer complètement une règle, dans la CloudWatch console, choisissez la règle que vous souhaitez supprimer, choisissez Actions, puis choisissez Supprimer.

Référence du format d'entrée des notifications de génération

CloudWatch fournit des notifications au format JSON.

Les notifications de changement d'état de génération utilisent le format suivant :

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "build-status": "SUCCEEDED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum": "6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-artifact.zip"
      }
    }
  },
}
```

```

"environment": {
  "image": "aws/codebuild/standard:5.0",
  "privileged-mode": false,
  "compute-type": "BUILD_GENERAL1_SMALL",
  "type": "LINUX_CONTAINER",
  "environment-variables": []
},
"timeout-in-minutes": 60,
"build-complete": true,
"initiator": "MyCodeBuildDemoUser",
"build-start-time": "Sep 1, 2017 4:12:29 PM",
"source": {
  "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
  "type": "S3"
},
"logs": {
  "group-name": "/aws/codebuild/my-sample-project",
  "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
  "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-edf953571bEX"
},
"phases": [
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:12:29 PM",
    "end-time": "Sep 1, 2017 4:12:29 PM",
    "duration-in-seconds": 0,
    "phase-type": "SUBMITTED",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:12:29 PM",
    "end-time": "Sep 1, 2017 4:13:05 PM",
    "duration-in-seconds": 36,
    "phase-type": "PROVISIONING",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:05 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 4,

```

```
    "phase-type": "DOWNLOAD_SOURCE",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "INSTALL",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "PRE_BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 70,
    "phase-type": "BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 0,
    "phase-type": "POST_BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 0,
    "phase-type": "UPLOAD_ARTIFACTS",
    "phase-status": "SUCCEEDED"
  },
  {
```

```

        "phase-context": [],
        "start-time": "Sep 1, 2017 4:14:21 PM",
        "end-time": "Sep 1, 2017 4:14:26 PM",
        "duration-in-seconds": 4,
        "phase-type": "FINALIZING",
        "phase-status": "SUCCEEDED"
    },
    {
        "start-time": "Sep 1, 2017 4:14:26 PM",
        "phase-type": "COMPLETED"
    }
]
},
"current-phase": "COMPLETED",
"current-phase-context": "[]",
"version": "1"
}
}

```

Les notifications de changement de phase de génération utilisent le format suivant :

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "completed-phase-context": "[]",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",

```



```

    "sha256sum":
      "6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
      "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
    },
    "environment": {
      "image": "aws/codebuild/standard:5.0",
      "privileged-mode": false,
      "compute-type": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environment-variables": []
    },
    "timeout-in-minutes": 60,
    "build-complete": true,
    "initiator": "MyCodeBuildDemoUser",
    "build-start-time": "Sep 1, 2017 4:12:29 PM",
    "source": {
      "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
      "type": "S3"
    },
    "logs": {
      "group-name": "/aws/codebuild/my-sample-project",
      "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
      "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
    },
    "phases": [
      {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:12:29 PM",
        "end-time": "Sep 1, 2017 4:12:29 PM",
        "duration-in-seconds": 0,
        "phase-type": "SUBMITTED",
        "phase-status": "SUCCEEDED"
      },
      {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:12:29 PM",
        "end-time": "Sep 1, 2017 4:13:05 PM",
        "duration-in-seconds": 36,
        "phase-type": "PROVISIONING",
        "phase-status": "SUCCEEDED"
      }
    ],

```

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:05 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 4,
  "phase-type": "DOWNLOAD_SOURCE",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
```

```
    "duration-in-seconds": 0,
    "phase-type": "UPLOAD_ARTIFACTS",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:26 PM",
    "duration-in-seconds": 4,
    "phase-type": "FINALIZING",
    "phase-status": "SUCCEEDED"
  },
  {
    "start-time": "Sep 1, 2017 4:14:26 PM",
    "phase-type": "COMPLETED"
  }
]
},
"completed-phase-status": "SUCCEEDED",
"completed-phase-duration-seconds": 4,
"version": "1",
"completed-phase-start": "Sep 1, 2017 4:14:21 PM",
"completed-phase-end": "Sep 1, 2017 4:14:26 PM"
}
```

Créez un échantillon de badges avec CodeBuild

AWS CodeBuild prend désormais en charge l'utilisation de badges de construction, qui fournissent une image intégrée et générée dynamiquement (badge) qui affiche l'état de la dernière version d'un projet. Cette image est accessible via une URL accessible au public générée pour votre CodeBuild projet. Cela permet à n'importe qui de consulter le statut d'un CodeBuild projet. Les badges de génération ne contiennent pas d'informations de sécurité. Ils n'ont donc pas besoin d'une authentification.

Rubriques

- [Créez un projet de construction avec des badges de construction](#)
- [Badges Access AWS CodeBuild Build](#)
- [Publier CodeBuild des badges de construction](#)
- [CodeBuild statuts des badges](#)

Créez un projet de construction avec des badges de construction

Utilisez l'une des procédures suivantes pour créer un projet de génération avec les badges de construction activés. Vous pouvez utiliser AWS CLI ou le AWS Management Console.

Pour créer un projet de construction avec les badges de construction activés (AWS CLI)

- Pour en savoir plus sur la création d'un projet de génération, consultez [Création d'un projet de génération \(AWS CLI\)](#). Pour inclure des badges de construction dans votre AWS CodeBuild projet, vous devez les spécifier *badgeEnabled* avec une valeur de `true`.

Pour créer un projet de construction avec les badges de construction activés (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, pour Fournisseur de source, choisissez le type de fournisseur de code source, puis effectuez l'une des actions suivantes :

Note

CodeBuild ne prend pas en charge les badges de génération auprès du fournisseur de source Amazon S3. Dans la AWS CodePipeline mesure où Amazon S3 est utilisé pour les transferts d'artefacts, les badges de construction ne sont pas pris en charge pour les projets de génération faisant partie d'un pipeline créé dans CodePipeline.

- Si vous avez choisi CodeCommit, pour Référentiel, sélectionnez le nom du référentiel. Sélectionnez Enable build badge (Activer le badge de génération) pour rendre le statut de génération de votre projet visible et intégrable.

- Si vous le souhaitez GitHub, suivez les instructions pour vous connecter (ou vous reconnecter) à GitHub. Sur la page GitHub Autoriser l'application, pour Accès à l'organisation, sélectionnez Demander l'accès à côté de chaque référentiel AWS CodeBuild auquel vous souhaitez pouvoir accéder. Après avoir choisi Authorize application, de retour dans la console AWS CodeBuild , pour Repository, choisissez le nom du référentiel qui contient le code source. Sélectionnez Enable build badge (Activer le badge de génération) pour rendre le statut de génération de votre projet visible et intégrable.
- Si vous avez choisi Bitbucket, suivez les instructions pour vous connecter (ou vous reconnecter) à Bitbucket. Dans la page Bitbucket Confirm access to your account, pour Organization access, choisissez Grant access. Après avoir choisi Autoriser l'accès, de retour dans la AWS CodeBuild console, pour Repository, choisissez le nom du référentiel qui contient le code source. Sélectionnez Enable build badge (Activer le badge de génération) pour rendre le statut de génération de votre projet visible et intégrable.

 Important

Si vous mettez à jour la source de votre projet, cela peut affecter la précision des badges de génération du projet.

5. Dans Environment (Environnement) :

Pour Image d'environnement, effectuez l'une des actions suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification

doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

6. Dans Rôle du service, sélectionnez l'une des options suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

 Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

7. Dans Buildspec, effectuez l'une des opérations suivantes :

- Choisissez Utiliser un fichier buildspec pour utiliser le fichier buildspec.yml dans le répertoire racine du code source.
- Choisissez Insérer des commandes de génération pour utiliser la console pour insérer des commandes de génération.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

8. Dans Artefacts, pour Type, effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.

- Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, `appspec.yml`, `target/my-app.jar`). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).
9. Développez Additional configuration (Configuration supplémentaire) et choisissez les options appropriées.
 10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Badges Access AWS CodeBuild Build

Vous pouvez utiliser AWS CodeBuild la console ou le AWS CLI pour accéder aux badges de build.

- Dans la CodeBuild console, dans la liste des projets de construction, dans la colonne Nom, choisissez le lien correspondant au projet de construction. Sur la *project-name* page Créer un projet :, dans Configuration, choisissez Copier l'URL du badge. Pour de plus amples informations, veuillez consulter [Affichage des détails d'un projet de génération \(console\)](#).
- Dans le AWS CLI, exécutez la `batch-get-projects` commande. L'URL de badge de génération est incluse dans la section des détails de l'environnement de projet de la sortie. Pour de plus amples informations, veuillez consulter [Affichage des détails d'un projet de génération \(AWS CLI\)](#).

L'URL de demande de badge de construction est générée avec une branche par défaut commune, mais vous pouvez spécifier n'importe quelle branche de votre référentiel source que vous avez utilisée pour exécuter une compilation. Par exemple :

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&branch=<branch>
```

Vous pouvez également spécifier une balise à partir de votre référentiel source en remplaçant le `branch` paramètre par le `tag` paramètre figurant dans l'URL du badge. Par exemple :

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&tag=<tag>
```

Publier CodeBuild des badges de construction

Vous pouvez afficher l'état de la dernière version dans un fichier Markdown en utilisant l'URL de votre badge de build dans une image Markdown. Cela est utile pour afficher l'état de la version la plus récente dans le fichier `readme.md` de votre référentiel source (par exemple, GitHub ou). CodeCommit Par exemple :

```

```

CodeBuild statuts des badges

Le badge de CodeBuild construction peut avoir l'un des statuts suivants.

- **PASSING** : La dernière génération sur la branche indiquée a abouti.
- **FAILING** : La dernière génération sur la branche indiquée a expiré, a échoué, comprend des erreurs ou a été arrêtée.
- **IN_PROGRESS** : La dernière génération sur la branche indiquée est en cours.
- **UNKNOWN** : Le projet n'a pas encore exécuté une génération pour la branche indiquée ou n'a exécuté aucune génération. En outre, la fonction de badges de génération a peut-être été désactivée.

« Rapport de test utilisant l'exemple AWS CLI »

Les tests que vous spécifiez dans votre fichier `buildspec` sont exécutés pendant votre génération. Cet exemple vous montre comment utiliser le AWS CLI pour intégrer des tests dans des builds in CodeBuild. Vous pouvez l'utiliser JUnit pour créer des tests unitaires, ou vous pouvez utiliser un autre outil pour créer des tests de configuration. Vous pouvez ensuite évaluer les résultats des tests pour résoudre les problèmes ou optimiser votre application.

Vous pouvez utiliser l' CodeBuild API ou la AWS CodeBuild console pour accéder aux résultats des tests. Cet exemple montre comment configurer votre rapport afin que ses résultats de test soient exportés vers un compartiment S3.

Rubriques

- [Exécutez l'exemple de rapport de test](#)

Exécutez l'exemple de rapport de test

Procédez comme suit pour exécuter l'exemple de rapport de test.

Rubriques

- [Conditions préalables](#)
- [Étape 1 : créer un groupe de rapports](#)
- [Étape 2 : Configuration d'un projet avec un groupe de rapports](#)
- [Étape 3 : Exécuter et afficher les résultats d'un rapport](#)

Conditions préalables

- Créez vos cas de test. Cet échantillon est rédigé en supposant que vous avez des cas de test à inclure dans votre rapport de test. Vous spécifiez l'emplacement de vos fichiers de test dans le fichier buildspec.

Les formats de fichier de rapport de test suivants sont pris en charge :

- Concombre JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- Visual Studio TRX XML (.xml)

Créez vos scénarios de test avec n'importe quel framework de test capable de créer des fichiers de rapport dans l'un de ces formats (par exemple, le JUnit plugin Surefire, TestNG ou Cucumber).

- Créez un compartiment S3 et notez son nom. Pour plus d'informations, consultez [Comment créer un compartiment S3 ?](#) dans le guide de l'utilisateur Amazon S3.
- Créez un rôle IAM et notez son ARN. Vous avez besoin de l'ARN lorsque vous créez votre projet de génération.
- Si votre rôle ne dispose pas des autorisations suivantes, ajoutez-les.

```
{  
    "Effect": "Allow",
```

```
"Resource": [
  "*"
],
"Action": [
  "codebuild:CreateReportGroup",
  "codebuild:CreateReport",
  "codebuild:UpdateReport",
  "codebuild:BatchPutTestCases"
]
}
```

Pour de plus amples informations, veuillez consulter [Autorisations pour les opérations de rapports de test](#).

Étape 1 : créer un groupe de rapports

1. Créez un fichier nommé `CreateReportGroupInput.json`.
2. Créez un dossier dans votre compartiment S3 où vos résultats de test sont exportés.
3. Copiez ce qui suit dans `CreateReportGroupInput.json`. Pour *<bucket-name>*, utilisez le nom du compartiment S3. Pour *<path-to-folder>*, entrez le chemin d'accès au dossier dans votre compartiment S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "path": "<path-to-folder>",
      "packaging": "NONE"
    }
  }
}
```

4. Exécutez la commande suivante dans le répertoire qui contient `CreateReportGroupInput.json`.

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

Le résultat se présente comme suit. Notez l'ARN pour `reportGroup`. Vous l'utilisez lorsque vous créez un projet qui utilise ce groupe de rapports.

```
{
  "reportGroup": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:report-group/<report-name>",
    "name": "<report-name>",
    "type": "TEST",
    "exportConfig": {
      "exportConfigType": "S3",
      "s3Destination": {
        "bucket": "<s3-bucket-name>",
        "path": "<folder-path>",
        "packaging": "NONE",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3"
      }
    },
    "created": 1570837165.885,
    "lastModified": 1570837165.885
  }
}
```

Étape 2 : Configuration d'un projet avec un groupe de rapports

Pour exécuter un rapport, vous devez d'abord CodeBuild créer un projet de génération configuré avec votre groupe de rapports. Les cas de test spécifiés pour votre groupe de rapports sont exécutés lorsque vous exécutez une génération.

1. Créez un fichier `buildspec.yml` nommé `buildspec.yml`.
2. Utilisez le YAML suivant comme modèle pour votre fichier `buildspec.yml`. Assurez-vous d'inclure les commandes qui exécutent vos tests. Dans la section `reports`, spécifiez les fichiers qui contiennent les résultats de vos cas de test. Ces fichiers stockent les résultats des tests auxquels vous pouvez accéder CodeBuild. Ils expirent 30 jours après leur création. Ces fichiers sont différents des fichiers de résultats de cas de test bruts que vous exportez vers un compartiment S3.

```
version: 0.2
phases:
install:
```

```

runtime-versions:
  java: openjdk8
build:
  commands:
    - echo Running tests
    - <enter commands to run your tests>

reports:
  <report-name-or-arn>: #test file information
  files:
    - '<test-result-files>'
  base-directory: '<optional-base-directory>'
  discard-paths: false #do not remove file paths from test result files

```

Note

Au lieu de l'ARN d'un groupe de rapports existant, vous pouvez également spécifier un nom pour un groupe de rapports qui n'a pas été créé. Si vous spécifiez un nom au lieu d'un ARN, CodeBuild crée un groupe de rapports lors de l'exécution d'une compilation. Son nom contient le nom de votre projet et le nom que vous spécifiez dans le fichier buildspec au format suivant : `project-name-report-group-name`. Pour plus d'informations, consultez [Création de rapports de test](#) et [Attribution des noms des groupes de rapports](#).

3. Créez un fichier nommé `project.json`. Ce fichier contient l'entrée de la commande `create-project`.
4. Copiez le JSON suivant dans `project.json`. Pour `source`, entrez le type et l'emplacement du référentiel qui contient vos fichiers source. Pour `serviceRole`, spécifiez l'ARN du rôle que vous utilisez.

```

{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|NO_SOURCE",
    "location": "<your-source-url>"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  }
}

```

```
{,
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "small"
  },
  "serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-name>"
}
```

5. Exécutez la commande suivante dans le répertoire qui contient `project.json`. Cela crée un projet nommé `test-project`.

```
aws codebuild create-project --cli-input-json file://project.json
```

Étape 3 : Exécuter et afficher les résultats d'un rapport

Dans cette section, vous exécutez une version du projet que vous avez créé précédemment. Au cours du processus de CodeBuild création, crée un rapport contenant les résultats des scénarios de test. Le rapport est contenu dans le groupe de rapports que vous avez spécifié.

1. Pour démarrer une compilation, exécutez la commande suivante. `test-report-project` est le nom du projet de construction créé ci-dessus. Notez l'ID de génération qui apparaît dans la sortie.

```
aws codebuild start-build --project-name test-report-project
```

2. Exécutez la commande suivante pour obtenir des informations sur votre génération, y compris l'ARN de votre rapport. Pour `<build-id>`, spécifiez votre ID de génération. Notez l'ARN du rapport dans la `reportArns` propriété de la sortie.

```
aws codebuild batch-get-builds --ids <build-id>
```

3. Exécutez la commande suivante pour obtenir des informations sur votre rapport. Pour `<report-arn>`, spécifiez votre ARN de rapport.

```
aws codebuild batch-get-reports --report-arns <report-arn>
```

Le résultat se présente comme suit. Cet exemple de sortie indique le nombre de tests réussis, échoués, ignorés, entraînés une erreur ou renvoyant un état inconnu.

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<amzn-s3-demo-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      },
      "expired": 1575916770.0,
      "truncated": false,
      "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",
      "type": "TEST",
      "arn": "<report-arn>",
      "testSummary": {
        "durationInNanoSeconds": 6657770,
        "total": 11,
        "statusCounts": {
          "FAILED": 3,
          "SKIPPED": 7,
          "ERROR": 0,
          "SUCCEEDED": 1,
          "UNKNOWN": 0
        }
      }
    }
  ],
  "reportsNotFound": []
}
```

```
}
```

4. Exécutez la commande suivante pour répertorier les informations sur les cas de test de votre rapport. Pour *<report-arn>*, spécifiez l'ARN de votre rapport. Pour le paramètre facultatif *--filter*, vous pouvez spécifier un résultat d'état (SUCCEEDED, FAILED, SKIPPED, ERROR ou UNKNOWN).

```
aws codebuild describe-test-cases \  
  --report-arn <report-arn> \  
  --filter status=SUCCEEDED|FAILED|SKIPPED|ERROR|UNKNOWN
```

Le résultat se présente comme suit.

```
{  
  "testCases": [  
    {  
      "status": "FAILED",  
      "name": "Test case 1",  
      "expired": 1575916770.0,  
      "reportArn": "<report-arn>",  
      "prefix": "Cucumber tests for agent",  
      "message": "A test message",  
      "durationInNanoSeconds": 1540540,  
      "testRawDataPath": "<path-to-output-report-files>"  
    },  
    {  
      "status": "SUCCEEDED",  
      "name": "Test case 2",  
      "expired": 1575916770.0,  
      "reportArn": "<report-arn>",  
      "prefix": "Cucumber tests for agent",  
      "message": "A test message",  
      "durationInNanoSeconds": 1540540,  
      "testRawDataPath": "<path-to-output-report-files>"  
    }  
  ]  
}
```

Exemples Docker pour CodeBuild

Cette section décrit des exemples d'intégrations entre Docker et AWS CodeBuild

Exemple	Description
Docker dans un exemple d'image personnalisée pour CodeBuild	Cet exemple crée et exécute une image Docker en utilisant une image CodeBuild de construction Docker personnalisée (<code>docker:dind</code> dans Docker Hub).
Exemple de serveur de génération d'images Docker pour CodeBuild	Cet exemple télécharge vos versions Docker vers un serveur de génération d'images géré.
Exemple de compilation Windows Docker pour CodeBuild	Cet exemple crée et exécute une image Docker Windows à l'aide CodeBuild de.
Exemple « Publier une image Docker dans un référentiel d'images Amazon ECR » pour CodeBuild	Cet exemple produit une image Docker en tant que sortie de compilation, puis envoie l'image Docker vers un référentiel d'images Amazon Elastic Container Registry (Amazon ECR).
Registre privé avec AWS Secrets Manager échantillon pour CodeBuild	Cet exemple montre comment utiliser une image Docker stockée dans un registre privé comme environnement d' CodeBuild exécution.

Docker dans un exemple d'image personnalisée pour CodeBuild

L'exemple suivant crée et exécute une image Docker à l'aide AWS CodeBuild d'une image de construction Docker personnalisée (`docker:dind` dans Docker Hub).

Pour savoir comment créer une image Docker en utilisant plutôt une image de construction fournie CodeBuild par le support Docker, consultez notre. [Exemple « Publier une image Docker sur Amazon ECR »](#)

⚠ Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [CodeBuild les sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatchTarification Amazon](#).

Rubriques

- [Exécutez le Docker dans un exemple d'image personnalisé](#)

Exécutez le Docker dans un exemple d'image personnalisé

Utilisez la procédure suivante pour exécuter le Docker dans un exemple d'image personnalisé. Pour plus d'informations sur cet exemple, consultez [Docker dans un exemple d'image personnalisé pour CodeBuild](#).

Pour exécuter le Docker dans un exemple d'image personnalisé

1. Créez les fichiers comme décrit dans les [Fichiers](#) sections [Structure de répertoire](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

⚠ Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

2. Créez un projet de build, exécutez le build et consultez les informations de build associées.

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerCustomImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "docker:dind",
    "computeType": "BUILD_GENERAL1_SMALL",
    "privilegedMode": false
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux Capabilities](#) sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

3. Pour voir les résultats de la génération, recherchez la chaîne Hello, World! dans le journal de la génération. Pour de plus amples informations, veuillez consulter [Affichage des détails d'une génération](#).

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Fichiers

L'image de base du système d'exploitation utilisé dans cet exemple est Ubuntu. L'exemple utilise ces fichiers.

`buildspec.yml` (dans *(root directory name)*)

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld echo "Hello, World!"
```

`Dockerfile` (dans *(root directory name)*)

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Exemple de serveur de génération d'images Docker pour CodeBuild

L'exemple suivant télécharge vos builds Docker vers un serveur de génération d'images géré. Vous pouvez adapter cet exemple pour fournir un serveur de génération d'images Docker dédié et géré dans la configuration de votre CodeBuild projet. Notez que l'instance provisionnée est active lorsque les builds sont exécutés activement pour le projet, et qu'elle est arrêtée lorsque les builds ne sont pas en cours d'exécution. L'instance provisionnée est stockée jusqu'à un mois avant d'être recyclée. Pour plus d'informations, consultez la section [CodeBuild Fonctionnalité du serveur Docker](#).

Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [CodeBuild les sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatchTarification Amazon](#).

Rubriques

- [Configuration d'un serveur Docker](#)

Configuration d'un serveur Docker

Utilisez la procédure suivante pour configurer un environnement informatique dédié à un CodeBuild projet qui gère les charges de travail Docker et stocke les couches d'images Docker.

Pour configurer un serveur Docker

1. Créez les fichiers comme décrit dans les [Fichiers](#) sections [Structure de répertoire](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

2. Créez un projet de build, exécutez le build et consultez les informations de build associées :
 - a. Dans la section Environnement de la console, choisissez Configuration supplémentaire, accédez à Configuration du serveur Docker, puis sélectionnez Activer le serveur docker pour ce projet. Vous pouvez ensuite choisir le type de calcul du serveur Docker et fournir un identifiant de registre.
 - b. Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la create-project commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
```

```

    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerServerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:5.0",
    "computeType": "BUILD_GENERAL1_LARGE",
    "dockerServer": [
      {
        "computeType": "BUILD_GENERAL1_LARGE",
        "securityGroupIds": [ "security-groups-ID" ]
      }
    ]
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}

```

Note

Les groupes de sécurité configurés pour les serveurs Docker doivent autoriser le trafic réseau entrant depuis le VPC configuré dans le projet. Ils devraient autoriser l'entrée sur le port 9876.

3. Pour voir les résultats de la génération, recherchez la chaîne Hello, World! dans le journal de la génération. Pour de plus amples informations, veuillez consulter [Affichage des détails d'une génération](#).

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

```

(root directory name)
### buildspec.yml
### Dockerfile

```

Fichiers

L'image de base du système d'exploitation utilisé dans cet exemple est Ubuntu. L'exemple utilise ces fichiers.

buildspec.yml (dans *(root directory name)*)

```
version: 0.2

phases:
  build:
    commands:
      - docker buildx build -t helloworld --load .
      - docker run helloworld echo "Hello, World!"
```

Dockerfile (dans *(root directory name)*)

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

RUN echo "Hello World"
```

Exemple de compilation Windows Docker pour CodeBuild

L'exemple suivant crée et exécute une image Docker Windows à l'aide CodeBuild de.

Rubriques

- [Exemple d'exécution de versions de Windows Docker](#)

Exemple d'exécution de versions de Windows Docker

Utilisez la procédure suivante pour exécuter les versions de Windows Docker.

Pour exécuter un exemple de build Windows Docker

1. Créez les fichiers comme décrit dans les [Fichiers](#) sections [Structure de répertoire](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

⚠ Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veuillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

2. Créez une WINDOWS_EC2 flotte.

Si vous utilisez le AWS CLI pour créer la flotte, l'entrée au format JSON de la `create-fleet` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "fleet-name",
  "baseCapacity": 1,
  "environmentType": "WINDOWS_EC2",
  "computeType": "BUILD_GENERAL1_MEDIUM"
}
```

3. Créez un projet de build, exécutez le build et consultez les informations de build associées.

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "project-name",
  "source": {
    "type": "S3",
    "location": "bucket-name/DockerImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "WINDOWS_EC2",
    "image": "Windows",
    "computeType": "BUILD_GENERAL1_MEDIUM",

```

```
"fleet": {
  "fleetArn": "fleet-arn"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name"
}
```

4. Pour voir les résultats de la génération, recherchez la chaîne Hello, World! dans le journal de la génération. Pour de plus amples informations, veuillez consulter [Affichage des détails d'une génération](#).

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Fichiers

L'image de base du système d'exploitation utilisé dans cet exemple est `mcr.microsoft.com/windows/servercore:ltsc2022`. L'exemple utilise ces fichiers.

`buildspec.yml` (dans (*root directory name*))

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld powershell -Command "Write-Host 'Hello World!'"
```

`Dockerfile` (dans (*root directory name*))

```
FROM mcr.microsoft.com/windows/servercore:ltsc2022
```



```
RUN powershell -Command "Write-Host 'Hello World'"
```

Exemple « Publier une image Docker dans un référentiel d'images Amazon ECR » pour CodeBuild

Cet exemple produit une image Docker en tant que sortie de compilation, puis envoie l'image Docker vers un référentiel d'images Amazon Elastic Container Registry (Amazon ECR). Vous pouvez adapter cet exemple pour transférer l'image Docker vers Docker Hub. Pour de plus amples informations, veuillez consulter [Adaptez l'exemple « Publier une image Docker sur Amazon ECR » pour le transférer vers Docker Hub](#).

Pour savoir comment générer une image Docker à l'aide d'une image de génération Docker personnalisée (docker:dind dans Docker Hub), consultez notre [Exemple Docker dans une image personnalisée](#).

Cet exemple a été testé en faisant référence à `golang:1.12`.

Cet exemple utilise la nouvelle fonction de générations Docker en plusieurs étapes, qui produit une image Docker comme sortie de génération. Il envoie ensuite l'image Docker vers un référentiel d'images Amazon ECR. Les générations d'image Docker en plusieurs étapes contribuent à réduire la taille de l'image Docker finale. Pour plus d'informations, consultez [Use multi-stage builds with Docker](#).

Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour AWS CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 AWS KMS, CloudWatch Logs et Amazon ECR. Pour plus d'informations, consultez [CodeBuild les rubriques Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) [Amazon et CloudWatch Tarification](#) [Amazon Elastic Container Registry](#).

Rubriques

- [Exécutez l'exemple « Publier une image Docker sur Amazon ECR »](#)
- [Adaptez l'exemple « Publier une image Docker sur Amazon ECR » pour le transférer vers Docker Hub](#)

Exécutez l'exemple « Publier une image Docker sur Amazon ECR »

Utilisez la procédure suivante pour exécuter l'exemple qui publie une image Docker sur Amazon ECR. Pour plus d'informations sur cet exemple, consultez. [Exemple « Publier une image Docker dans un référentiel d'images Amazon ECR » pour CodeBuild](#)

Pour exécuter cet exemple

1. Si vous avez déjà un référentiel d'images dans Amazon ECR que vous souhaitez utiliser, passez à l'étape 3. Sinon, si vous utilisez un utilisateur au lieu d'un compte AWS root ou un utilisateur administrateur pour travailler avec Amazon ECR, ajoutez cette déclaration (entre **### BEGIN ADDING STATEMENT HERE ###** et **### END ADDING STATEMENT HERE ###**) à l'utilisateur (ou au groupe IAM auquel l'utilisateur est associé). L'utilisation d'un compte AWS root n'est pas recommandée. Cette déclaration permet la création de référentiels Amazon ECR pour le stockage d'images Docker. Des ellipses (. . .) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter l'instruction. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie. Pour plus d'informations, consultez la section [Utilisation des politiques intégrées à l'aide du AWS Management Console](#) guide de l'utilisateur.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

L'entité IAM qui modifie cette politique doit être autorisée dans IAM à modifier les politiques.

2. Créez un référentiel d'images dans Amazon ECR. Assurez-vous de créer le référentiel dans la même AWS région que celle dans laquelle vous créez votre environnement de génération et exécutez votre build. Pour plus d'informations, consultez la section [Création d'un référentiel](#) dans le guide de l'utilisateur Amazon ECR. Le nom de ce référentiel doit correspondre au nom du référentiel que vous spécifierez ultérieurement dans cette procédure, représenté par la variable d'environnement IMAGE_REPO_NAME. Assurez-vous que la politique de référentiel Amazon ECR accorde un accès push aux images pour votre rôle IAM CodeBuild de service.
3. Ajoutez cette déclaration (entre **### BEGIN ADDING STATEMENT HERE ###** et **### END ADDING STATEMENT HERE ###**) à la politique que vous avez attachée à votre rôle AWS CodeBuild de service. Cette instruction permet de CodeBuild télécharger des images Docker dans les référentiels Amazon ECR. Des ellipses (. . .) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter l'instruction. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

L'entité IAM qui modifie cette politique doit être autorisée dans IAM à modifier les politiques.

4. Créez les fichiers comme décrit dans les [Fichiers](#) sections [Structure de répertoire](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub Pour plus d'informations, reportez-vous à la section [Référence des fichiers de définitions d'images](#) dans le Guide de AWS CodePipeline l'utilisateur.

 Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

5. Créez un projet de build, exécutez le build et visualisez les informations de build.

Si vous utilisez la console pour créer votre projet :

- a. Pour Système d'exploitation, choisissez Ubuntu.
- b. Pour Runtime (Exécution), sélectionnez Standard.
- c. Pour Image, choisissez aws/codebuild/standard : 5.0.
- d. Ajoutez les variables d'environnement suivantes :
 - AWS_DEFAULT_REGION avec une valeur de *region-ID*
 - AWS_ACCOUNT_ID avec une valeur de *account-ID*
 - IMAGE_TAG avec la valeur Latest
 - IMAGE_REPO_NAME avec une valeur de *Amazon-ECR-repo-name*

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "sample-docker-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      },
      {
        "name": "AWS_ACCOUNT_ID",
        "value": "account-ID"
      },
      {
        "name": "IMAGE_REPO_NAME",
        "value": "Amazon-ECR-repo-name"
      },
      {
        "name": "IMAGE_TAG",
        "value": "latest"
      }
    ],
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

6. Confirmez que CodeBuild l'image Docker a bien été transférée vers le référentiel :

1. Ouvrez la console Amazon ECR à <https://console.aws.amazon.com/ecr/> l'adresse.
2. Choisissez le nom du référentiel. L'image doit apparaître dans la colonne Balise d'image.

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Fichiers

Cet exemple utilise ces fichiers.

buildspec.yml (dans *(root directory name)*)

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
    build:
      commands:
        - echo Build started on `date`
        - echo Building the Docker image...
        - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
        - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
    post_build:
      commands:
        - echo Build completed on `date`
        - echo Pushing the Docker image...
        - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
```

Dockerfile (dans *(root directory name)*)

```
FROM golang:1.12-alpine AS build
#Install git
RUN apk add --no-cache git
#Get the hello world package from a GitHub repository
RUN go get github.com/golang/example/hello
WORKDIR /go/src/github.com/golang/example/hello
# Build the project and send the output to /bin/HelloWorld
RUN go build -o /bin/HelloWorld

FROM golang:1.12-alpine
#Copy the build's output binary from the previous build container
COPY --from=build /bin/HelloWorld /bin/HelloWorld
ENTRYPOINT ["/bin/HelloWorld"]
```

 Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Adaptez l'exemple « Publier une image Docker sur Amazon ECR » pour le transférer vers Docker Hub

Pour adapter l'exemple « Publier une image Docker sur Amazon ECR » afin que l'image Docker soit envoyée vers Docker Hub plutôt que vers Amazon ECR, modifiez le code de l'exemple. Pour plus d'informations sur l'exemple, reportez-vous aux sections [Exemple « Publier une image Docker dans un référentiel d'images Amazon ECR » pour CodeBuild](#) et [Exécutez l'exemple « Publier une image Docker sur Amazon ECR »](#).

 Note

Si vous utilisez une version de Docker antérieure à 17.06, supprimez l'option `--no-include-email`.

1. Remplacez ces lignes de code spécifiques à Amazon ECR dans le `buildspec.yml` fichier :

```
...
pre_build:
  commands:
```

```

- echo Logging in to Amazon ECR...
- aws ecr get-login-password --region $AWS_DEFAULT_REGION |
docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
...

```

Par ces lignes de code spécifiques à Docker Hub :

```

...
pre_build:
  commands:
    - echo Logging in to Docker Hub...
    # Type the command to log in to your Docker Hub account here.
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $IMAGE_REPO_NAME:$IMAGE_TAG
...

```

2. Téléchargez le code modifié dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

⚠ Important

Ne chargez pas (*root directory name*), mais seulement les fichiers à l'intérieur de (*root directory name*).

Si vous utilisez un compartiment d'entrée S3, veuillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas (*root directory name*) au fichier ZIP, mais seulement les fichiers à l'intérieur de (*root directory name*).

3. Remplacez ces lignes de code de l'entrée au format JSON vers la commande `create-project` :

```
...
  "environmentVariables": [
    {
      "name": "AWS_DEFAULT_REGION",
      "value": "region-ID"
    },
    {
      "name": "AWS_ACCOUNT_ID",
      "value": "account-ID"
    },
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ]
...
```

Par ces lignes de code :

```
...
  "environmentVariables": [
    {
      "name": "IMAGE_REPO_NAME",
      "value": "your-Docker-Hub-repo-name"
    }
  ]
...
```

```
    },  
    {  
      "name": "IMAGE_TAG",  
      "value": "latest"  
    }  
  ]  
  ...
```

4. Créez un environnement de génération, exécutez le build et consultez les informations de build associées.
5. Vérifiez que l'image Docker a AWS CodeBuild bien été transférée vers le référentiel. Connectez-vous à Docker Hub, accédez au référentiel et choisissez l'onglet Tags. La balise `latest` doit comporter une valeur Last Updated très récente.

Registre privé avec AWS Secrets Manager échantillon pour CodeBuild

Cet exemple vous montre comment utiliser une image Docker stockée dans un registre privé comme environnement AWS CodeBuild d'exécution. Les informations d'identification du registre privé sont stockées dans AWS Secrets Manager. N'importe quel registre privé fonctionne avec CodeBuild. Cet exemple utilise Docker Hub.

Note

Les secrets sont visibles par les actions et ne sont pas masqués lorsqu'ils sont écrits dans un fichier.

Rubriques

- [Exigences de l'exemple de registre privé](#)
- [Création d'un CodeBuild projet avec un registre privé](#)
- [Configurer un identifiant de registre privé pour les coureurs auto-hébergés](#)

Exigences de l'exemple de registre privé

Pour utiliser un registre privé avec AWS CodeBuild, vous devez disposer des éléments suivants :

- Un secret du Gestionnaire de Secrets qui stocke vos informations d'identification Docker Hub. Les informations d'identification sont utilisées pour accéder à votre référentiel privé.

 Note

Les secrets que vous créez vous seront facturés.

- Un référentiel ou un compte privé.
- Une politique IAM de rôle de CodeBuild service qui accorde l'accès à votre secret Secrets Manager.

Suivez ces étapes pour créer ces ressources, puis créez un projet de CodeBuild construction à l'aide des images Docker stockées dans votre registre privé.

Création d'un CodeBuild projet avec un registre privé

1. Pour plus d'informations sur la création d'un référentiel privé, consultez [Référentiels sur Docker Hub](#). Vous pouvez également exécuter les commandes suivantes dans un terminal pour extraire une image, obtenir son ID et la transmettre à un nouveau référentiel.

```
docker pull amazonlinux
docker images amazonlinux --format {{.ID}}
docker tag image-id your-username/repository-name:tag
docker login
docker push your-username/repository-name
```

2. Suivez les étapes décrites dans la [section Créer un AWS Secrets Manager secret](#) dans le guide de AWS Secrets Manager l'utilisateur.
 - a. À l'étape 3, dans Choisir le type de secret, choisissez Autre type de secret.
 - b. Dans les paires clé/valeur, créez une paire clé-valeur pour votre nom d'utilisateur Docker Hub et une paire clé-valeur pour votre mot de passe Docker Hub.
 - c. Continuez à suivre les étapes décrites dans [Créer un AWS Secrets Manager secret](#).
 - d. À l'étape 5, sur la page Configurer la rotation automatique, désactivez-la car les clés correspondent à vos informations d'identification Docker Hub.
 - e. Terminez en suivant les étapes décrites dans [Créer un AWS Secrets Manager secret](#).

Pour plus d'informations, consultez [Qu'est-ce qu' AWS Secrets Manager ?](#)

3. Lorsque vous créez un AWS CodeBuild projet dans la console, il CodeBuild joint l'autorisation requise pour vous. Si vous utilisez une AWS KMS clé autre que `DefaultEncryptionKey`, vous devez l'ajouter au rôle de service. Pour plus d'informations, consultez la section [Modification d'un rôle \(console\)](#) dans le guide de l'utilisateur IAM.

Pour que votre rôle de service fonctionne avec Secrets Manager, il doit au minimum disposer de l'`secretsmanager:GetSecretValue` autorisation.

▼ KMS (1 action) Clone Remove

Service KMS

Actions Write
Decrypt

Resources ☒ Specific close ☐ All resources

key EDIT ☐ Any

[Add ARN to restrict access](#)

Request conditions [Specify request conditions \(optional\)](#)

4. Pour utiliser la console afin de créer un projet avec un environnement stocké dans un registre privé, effectuez les opérations suivantes lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

Note

Si votre registre privé se trouve dans votre VPC, celui-ci doit disposer d'un accès public à Internet. CodeBuild Impossible d'extraire une image d'une adresse IP privée dans un VPC.

- a. Dans Image d'environnement, sélectionnez Image personnalisée.
- b. Pour Type d'environnement, choisissez Linux ou Windows.
- c. Pour le registre d'images, choisissez Autre registre.
- d. Dans URL de registre externe, entrez l'emplacement de l'image et dans Informations d'identification du registre. Entrez éventuellement l'ARN ou le nom de vos informations d'identification de Secrets Manager.

Note

Si vos informations d'identification n'existent pas dans votre région actuelle, vous devez utiliser l'ARN. Vous ne pouvez pas utiliser le nom des informations d'identification si celles-ci existent dans une autre région.

Configurer un identifiant de registre privé pour les coureurs auto-hébergés

Suivez les instructions ci-dessous pour configurer un identifiant de registre pour un coureur auto-hébergé.

Note

Notez que ces informations d'identification ne seront utilisées que si les images sont remplacées par celles provenant de registres privés.

AWS Management Console

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de construction ou sélectionnez un projet existant. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Modification des paramètres d'un projet de génération \(console\)](#).
3. Dans Environnement, sélectionnez Configuration supplémentaire.
4. Dans Configuration supplémentaire, entrez le nom ou l'ARN du formulaire secret AWS Secrets Manager pour les informations d'identification du registre (facultatif).

Registry credential - *optional*

AWS CLI

1. Si vous souhaitez créer un nouveau projet, exécutez la commande create-project.

```
aws codebuild create-project \
```

```
--name project-name \  
--source type=source-type,location=source-location \  
--environment "type=environment-type,image=image,computeType=compute-  
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-  
name-or-arn},imagePullCredentialsType=CODEBUILD|SERVICE_ROLE" \  
--artifacts type=artifacts-type \  
--service-role arn:aws:iam::account-ID:role/service-role/service-role-name
```

2. Si vous souhaitez mettre à jour un projet existant, exécutez la commande `update-project`.

```
aws codebuild update-project \  
--name project-name \  
--environment "type=environment-type,image=image,computeType=compute-  
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-  
name-or-arn}"
```

Création d'un site web statique avec la sortie de génération hébergée dans un compartiment S3

Vous pouvez désactiver le chiffrement des artefacts dans une génération. Cela peut s'avérer utile pour publier des artefacts dans un emplacement configuré pour héberger un site web. (Vous ne pouvez pas publier d'artefacts chiffrés.) Cet exemple montre comment utiliser des webhooks pour déclencher une génération et publier ses artefacts dans un compartiment S3 qui est configuré comme un site web.

1. Suivez les instructions indiquées dans [Configuration d'un site web statique](#) pour configurer un compartiment S3 afin qu'il fonctionne comme un site web.
2. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
3. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
4. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
5. Dans Source, pour Fournisseur de source, choisissez GitHub. Suivez les instructions pour vous connecter (ou vous reconnecter) à GitHub, puis choisissez Autoriser.

Pour Webhook, sélectionnez Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel. Vous pouvez cocher cette case uniquement si vous avez choisi Use a repository in my account (Utiliser un référentiel de mon compte).

SourceAdd source

Source 1 - Primary

Source provider

GitHub

Repository

☐ Public repository

☒ Repository in my GitHub account

GitHub repository

Disconnect GitHub account

▼ Additional configuration

Git clone depth

Git clone depth - optional

1

Build Status - optional

☐ Report build statuses to source provider when your builds start and finish

Webhook - optional

☒ Rebuild every time a code change is pushed to this repository

Branch filter - optional

Enter a regular expression

6. Dans Environment (Environnement) :

Pour Image d'environnement, effectuez l'une des actions suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
 - Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
 - Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .
7. Dans Rôle du service, sélectionnez l'une des options suivantes :
- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
 - Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

 Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

8. Dans Buildspec, effectuez l'une des opérations suivantes :
- Choisissez Utiliser un fichier buildspec pour utiliser le fichier buildspec.yml dans le répertoire racine du code source.

- Choisissez Insérer des commandes de génération pour utiliser la console pour insérer des commandes de génération.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

9. Dans Artifacts, pour Type, choisissez Amazon S3 pour stocker la sortie de compilation dans un compartiment S3.
10. Pour Bucket name (Nom du compartiment), choisissez le nom du compartiment S3 que vous avez configuré pour fonctionner comme un site web à l'étape 1.
11. Si vous avez choisi Insérer des commandes de génération précédemment dans la procédure Environnement, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le compartiment de sortie. Si vous avez plusieurs emplacements, utilisez une virgule pour séparer chaque emplacement (par exemple, **appspec.yml, target/my-app.jar**). Pour de plus amples informations, veuillez consulter [Artifacts reference-key in the buildspec file](#).
12. Sélectionnez Disable artifacts encryption (Désactiver le chiffrement des artefacts).
13. Développez Additional configuration (Configuration supplémentaire) et choisissez les options appropriées.
14. Choisissez Créer un projet de génération. Sur la page du projet de génération, dans Historique de génération, choisissez Démarrer la génération pour exécuter la génération.
15. (Facultatif) Suivez les instructions de la [section Exemple : accélérez votre site Web avec Amazon CloudFront](#) dans le manuel du développeur Amazon S3.

Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie

Vous pouvez créer un projet de AWS CodeBuild construction avec plusieurs sources d'entrée et plusieurs ensembles d'artefacts de sortie. Cet exemple vous montre comment configurer un projet de génération qui :

- utilise plusieurs sources et référentiels de différents types ;
- publie des artefacts de génération vers plusieurs compartiments S3 dans une seule génération.

Dans l'exemple suivant, vous allez créer un projet de build et l'utiliser pour exécuter une build. L'exemple utilise le fichier buildspec du projet de génération pour vous montrer comment intégrer plusieurs sources et créer plusieurs ensembles d'artefacts.

Pour savoir comment créer un pipeline qui utilise plusieurs entrées source afin de CodeBuild créer plusieurs artefacts de sortie, voir [Exemple d' CodePipeline/CodeBuild intégration avec plusieurs sources d'entrée et artefacts de sortie](#).

Rubriques

- [Créez un projet de construction avec plusieurs entrées et sorties](#)
- [Création d'un projet de construction sans source](#)

Créez un projet de construction avec plusieurs entrées et sorties

Utilisez la procédure suivante pour créer un projet de construction avec plusieurs entrées et sorties.

Pour créer un projet de construction avec plusieurs entrées et sorties

1. Téléchargez vos sources vers un ou plusieurs buckets S3, CodeCommit GitHub, GitHub Enterprise Server ou référentiels Bitbucket.
2. Choisissez quelle source est la source principale. Il s'agit de la source dans laquelle vous CodeBuild recherchez et exécutez votre fichier buildspec.
3. Créez un projet de génération. Pour de plus amples informations, veuillez consulter [Créez un projet de construction dans AWS CodeBuild](#).
4. Créez votre projet de build, exécutez le build et obtenez des informations sur le build.
5. Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ce qui suit :

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
```

```
    "sourceIdentifier": "source1"
  },
  {
    "type": "GITHUB",
    "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
    "sourceIdentifier": "source2"
  }
],
"secondaryArtifacts": [ss
  {
    "type": "S3",
    "location": "<output-bucket>",
    "artifactIdentifier": "artifact1"
  },
  {
    "type": "S3",
    "location": "<other-output-bucket>",
    "artifactIdentifier": "artifact2"
  }
],
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Votre source principale est définie sous l'attribut `source`. Toutes les autres sources sont appelées « sources secondaires » et apparaissent sous `secondarySources`. Toutes les sources secondaires sont installées dans leur propre répertoire. Ce répertoire est stocké dans la variable d'environnement intégrée `CODEBUILD_SRC_DIR_<sourceIdentifier>`. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

L'attribut `secondaryArtifacts` contient une liste de définitions d'artefacts. Ces artefacts utilisent le bloc `secondary-artifacts` du fichier `buildspec` qui est imbriqué dans le bloc `artifacts`.

Les artefacts secondaires du fichier `buildspec` ont la même structure que tous les artefacts et sont séparés par leur identifiant d'artefact.

 Note

Dans l'API [CodeBuild](#), l'attribut `artifactIdentifier` sur un artefact secondaire est obligatoire dans `CreateProject` et `UpdateProject`. Il doit être utilisé pour référencer un artefact secondaire.

Lorsque vous utilisez l'entrée précédente au format JSON, le fichier `buildspec` du projet peut ressembler à ce qui suit :

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk11
  build:
    commands:
      - cd $CODEBUILD_SRC_DIR_source1
      - touch file1
      - cd $CODEBUILD_SRC_DIR_source2
      - touch file2

artifacts:
  files:
    - '**.*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR_source1
      files:
        - file1
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - file2
```

Vous pouvez substituer la version de la source principale à l'aide de l'API avec l'attribut `sourceVersion` dans `StartBuild`. Pour substituer une ou plusieurs versions de source secondaire, utilisez l'attribut `secondarySourceVersionOverride`.

L'entrée au format JSON de la `start-build` commande dans le AWS CLI peut ressembler à ceci :

```
{
  "projectName": "sample-project",
  "secondarySourcesVersionOverride": [
    {
      "sourceIdentifier": "source1",
      "sourceVersion": "codecommit-branch"
    },
    {
      "sourceIdentifier": "source2",
      "sourceVersion": "github-branch"
    },
  ],
}
```

Création d'un projet de construction sans source

Vous pouvez configurer un CodeBuild projet en choisissant le type de **NO_SOURCE** source lorsque vous configurez votre source. Lorsque votre type de source est **NO_SOURCE**, vous ne pouvez pas spécifier de fichier buildspec car votre projet n'a pas de source. Au lieu de cela, vous devez spécifier une chaîne buildspec au format YAML dans l'attribut buildspec de l'entrée au format JSON pour la commande de l'interface de ligne de commande `create-project`. Elle peut ressembler à ceci :

```
{
  "name": "project-name",
  "source": {
    "type": "NO_SOURCE",
    "buildspec": "version: 0.2\n\nphases:\n  build:\n    commands:\n      - command"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

Versions d'exécution dans un exemple de fichier buildspec pour CodeBuild

Si vous utilisez l'image standard Amazon Linux 2 (AL2) version 1.0 ou ultérieure, ou l'image standard Ubuntu version 2.0 ou ultérieure, vous pouvez spécifier un ou plusieurs environnements d'exécution dans la `runtime-versions` section de votre fichier buildspec. Les exemples suivants montrent comment modifier le temps d'exécution de votre projet, spécifier plusieurs environnements d'exécution et spécifier un environnement d'exécution dépendant d'un autre environnement d'exécution. Pour plus d'informations sur les environnements d'exécution pris en charge, consultez [Images Docker fournies par CodeBuild](#).

Note

Si vous utilisez Docker dans le conteneur de générations, la génération doit s'exécuter en mode privilégié. Pour plus d'informations, consultez [Exécuter AWS CodeBuild les builds manuellement](#) et [Créez un projet de construction dans AWS CodeBuild](#).

Rubriques

- [Mettre à jour la version d'exécution dans le fichier buildspec](#)
- [Spécification de deux environnements d'exécution](#)

Mettre à jour la version d'exécution dans le fichier buildspec

Vous pouvez modifier le runtime utilisé par votre projet vers une nouvelle version en mettant à jour la `runtime-versions` section de votre fichier buildspec. Les exemples suivants montrent comment spécifier Java versions 8 et 11 :

- Une section `runtime-versions` qui spécifie la version 8 de Java :

```
phases:
  install:
    runtime-versions:
      java: corretto8
```

- Une section `runtime-versions` qui spécifie la version 11 de Java :

```
phases:
  install:
    runtime-versions:
      java: corretto11
```

Les exemples suivants montrent comment spécifier différentes versions de Python à l'aide de l'image standard d'Ubuntu 5.0 ou de l'image standard 3.0 d'Amazon Linux 2 :

- `runtime-versions`Section qui spécifie la version 3.7 de Python :

```
phases:
  install:
    runtime-versions:
      python: 3.7
```

- `runtime-versions`Section qui spécifie la version 3.8 de Python :

```
phases:
  install:
    runtime-versions:
      python: 3.8
```

Cet exemple illustre un projet qui démarre avec l'environnement d'exécution Java version 8, puis qui est mis à jour vers l'environnement d'exécution Java version 10.

1. Téléchargez et installez Maven. Pour de plus amples informations, consultez [Downloading Apache Maven](#) et [Installing Apache Maven](#) sur le site Web Apache Maven.
2. Placez-vous dans un répertoire vide sur votre ordinateur ou instance local, puis exécutez cette commande Maven.

```
mvn archetype:generate "-DgroupId=com.mycompany.app" "-DartifactId=R00T" "-DarchetypeArtifactId=maven-archetype-webapp" "-DinteractiveMode=false"
```

En cas de réussite, cette structure de répertoire et ces fichiers sont créés.

```
.
### R00T
```

```
### pom.xml
### src
  ### main
    ### resources
    ### webapp
      ### WEB-INF
      #   ### web.xml
      ### index.jsp
```

3. Créez un fichier nommé `buildspec.yml` avec les contenus suivants. Stockez le fichier dans le répertoire *(root directory name)/my-web-app*.

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  build:
    commands:
      - java -version
      - mvn package
artifacts:
  files:
    - '**/*'
  base-directory: 'target/my-web-app'
```

Dans le fichier `buildspec` :

- La section `runtime-versions` spécifie que le projet utilise la version 8 de l'environnement d'exécution Java.
- La commande `- java -version` affiche la version de Java utilisée par votre projet lors de la génération.

La structure de votre fichier devrait maintenant ressembler à ceci.

```
(root directory name)
### my-web-app
  ### src
  #   ### main
  #   ### resources
```



```
#   ### webapp
#       ### WEB-INF
#           ### web.xml
#               ### index.jsp
### buildspec.yml
### pom.xml
```

4. Téléchargez le contenu du my-web-app répertoire dans un compartiment d'entrée S3 ou un CodeCommit référentiel Bitbucket. GitHub

 Important

Ne chargez pas *(root directory name)* ou *(root directory name)/my-web-app*, mais seulement les répertoires et les fichiers de *(root directory name)/my-web-app*.

Si vous utilisez un compartiment d'entrée S3, veuillez à créer un fichier ZIP qui contient la structure de répertoire et les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* ou *(root directory name)/my-web-app* au fichier ZIP, mais seulement les répertoires et les fichiers de *(root directory name)/my-web-app*.

5. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
6. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#). Laissez tous les paramètres avec leurs valeurs par défaut, sauf pour ces paramètres.
 - Pour Environnement :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez aws/codebuild/amazonlinux-x86_64-standard:4.0.
7. Choisissez Démarrer la génération.
8. Dans Configuration de la génération, acceptez les valeurs par défaut, puis choisissez Démarrer la génération.
9. Une fois que la génération est terminée, affichez sa sortie dans l'onglet Journaux de génération. Vous devez voir des résultats similaires à ce qui suit :

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto8' based on manual
selections...
[Container] Date Time Running command echo "Installing Java version 8 ..."
Installing Java version 8 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_8_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_8_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_8_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_8_HOME"/bin/*
"$JRE_8_HOME"/bin/*;
```

10. Mise à jour de la section runtime-versions avec Java version 11 :

```
install:
  runtime-versions:
    java: corretto11
```

11. Une fois que vous avez enregistré la modification, réexécutez votre génération et affichez sa sortie. Vous devez voir que la version installée de Java est la version 11. Vous devez voir des résultats similaires à ce qui suit :

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual
selections...
Installing Java version 11 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"
```

```
[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"
```

```
[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*  
"$JRE_11_HOME"/bin/*;
```

Spécification de deux environnements d'exécution

Vous pouvez spécifier plusieurs environnements d'exécution dans le même projet de CodeBuild construction. Cet exemple de projet utilise deux fichiers sources : l'un qui utilise l'environnement d'exécution Go et l'autre qui utilise l'environnement d'exécution Node.js.

1. Créez un répertoire nommé `my-source`.
2. Dans le répertoire `my-source`, créez un répertoire nommé `golang-app`.
3. Créez un fichier nommé `hello.go` avec les contenus suivants. Stockez le fichier dans le répertoire `golang-app`.

```
package main  
import "fmt"  
  
func main() {  
    fmt.Println("hello world from golang")  
    fmt.Println("1+1 =", 1+1)  
    fmt.Println("7.0/3.0 =", 7.0/3.0)  
    fmt.Println(true && false)  
    fmt.Println(true || false)  
    fmt.Println(!true)  
    fmt.Println("good bye from golang")  
}
```

4. Dans le répertoire `my-source`, créez un répertoire nommé `nodejs-app`. Il doit être au même niveau que le répertoire `golang-app`.
5. Créez un fichier nommé `index.js` avec les contenus suivants. Stockez le fichier dans le répertoire `nodejs-app`.

```
console.log("hello world from nodejs");  
console.log("1+1 =" + (1+1));  
console.log("7.0/3.0 =" + 7.0/3.0);  
console.log(true && false);  
console.log(true || false);
```

```
console.log(!true);
console.log("good bye from nodejs");
```

6. Créez un fichier nommé `package.json` avec les contenus suivants. Stockez le fichier dans le répertoire `nodejs-app`.

```
{
  "name": "mycompany-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"run some tests here\""
  },
  "author": "",
  "license": "ISC"
}
```

7. Créez un fichier nommé `buildspec.yml` avec les contenus suivants. Stockez le fichier dans le répertoire `my-source`, au même niveau que les répertoires `nodejs-app` et `golang-app`. La `runtime-versions` section spécifie les environnements d'exécution de Node.js version 12 et Go version 1.13.

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
      nodejs: 12
  build:
    commands:
      - echo Building the Go code...
      - cd $CODEBUILD_SRC_DIR/golang-app
      - go build hello.go
      - echo Building the Node code...
      - cd $CODEBUILD_SRC_DIR/nodejs-app
      - npm run test
  artifacts:
    secondary-artifacts:
      golang_artifacts:
        base-directory: golang-app
        files:
```

```
- hello
nodejs_artifacts:
  base-directory: nodejs-app
  files:
    - index.js
    - package.json
```

8. La structure de votre fichier devrait maintenant ressembler à ceci.

```
my-source
### golang-app
#   ### hello.go
### nodejs.app
#   ### index.js
#   ### package.json
### buildspect.yml
```

9. Téléchargez le contenu du my-source répertoire dans un compartiment d'entrée S3 ou un CodeCommit référentiel Bitbucket. GitHub

 Important

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP qui contient la structure de répertoire et les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas my-source au fichier ZIP, mais seulement les répertoires et les fichiers dans my-source.

10. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
11. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#). Laissez tous les paramètres avec leurs valeurs par défaut, sauf pour ces paramètres.

- Pour Environnement :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez aws/codebuild/amazonlinux-x86_64-standard:4.0.

12. Choisissez Créer un projet de génération.

13. Choisissez Démarrer la génération.
14. Dans Configuration de la génération, acceptez les valeurs par défaut, puis choisissez Démarrer la génération.
15. Une fois que la génération est terminée, affichez sa sortie dans l'onglet Journaux de génération. Vous devez visualiser des résultats similaires à ce qui suit. Ils montrent la sortie des environnements d'exécution Go et Node.js. Ils montrent également la sortie des applications Go et Node.js.

```
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'golang' runtime version '1.13' based on manual
selections...
[Container] Date Time Selecting 'nodejs' runtime version '12' based on manual
selections...
[Container] Date Time Running command echo "Installing Go version 1.13 ..."
Installing Go version 1.13 ...

[Container] Date Time Running command echo "Installing Node.js version 12 ..."
Installing Node.js version 12 ...

[Container] Date Time Running command n $NODE_12_VERSION
installed : v12.20.1 (with npm 6.14.10)

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time  INSTALL: 0 commands
[Container] Date Time  BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/golang-app

[Container] Date Time Running command go build hello.go
```

```
[Container] Date Time Running command echo Building the Node code...
Building the Node code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/nodejs-app

[Container] Date Time Running command npm run test

> mycompany-app@1.0.0 test /codebuild/output/src924084119/src/nodejs-app
> echo "run some tests here"

run some tests here
```

Exemple de version source avec AWS CodeBuild

Cet exemple illustre la façon de spécifier une version de votre code source à l'aide d'un format autre qu'un ID de validation (également connu sous le nom de SHA de validation). Vous pouvez spécifier la version de votre source de différentes manières :

- Pour un fournisseur de source Amazon S3, utilisez l'ID de version de l'objet qui représente le fichier ZIP d'entrée de compilation.
- Pour CodeCommit Bitbucket et GitHub Enterprise Server GitHub, utilisez l'une des options suivantes :
 - Demande d'extraction en tant que référence de demande d'extraction (par exemple, `refs/pull/1/head`).
 - Branche en tant que nom de branche.
 - ID de validation.
 - Balise.
 - Référence et identifiant de validation. La référence peut être l'une des suivantes :
 - Une balise (par exemple, `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Une branche (par exemple, `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Une demande d'extraction (par exemple, `refs/pull/1/head^{full-commit-SHA}`).
- Pour GitLab et GitLab Autogéré, utilisez l'une des options suivantes :
 - Branche en tant que nom de branche.
 - ID de validation.

- Balise.

Note

Vous pouvez spécifier la version d'une source de pull request uniquement si votre référentiel est GitHub GitHub Enterprise Server.

Si vous utilisez une référence et un ID de validation pour spécifier une version, la phase `DOWNLOAD_SOURCE` de la génération est plus rapide que si vous indiquez uniquement la version. En effet, lorsque vous ajoutez une référence, il CodeBuild n'est pas nécessaire de télécharger l'intégralité du référentiel pour trouver le commit.

- Vous pouvez spécifier une version source avec seulement un ID de validation, par exemple `12345678901234567890123467890123456789`. Dans ce cas, vous CodeBuild devez télécharger l'intégralité du référentiel pour trouver la version.
- Vous pouvez spécifier une version source avec une référence et un ID de validation dans ce format : `refs/heads/branchname^{full-commit-SHA}` (par exemple, `refs/heads/main^{12345678901234567890123467890123456789}`). Dans ce cas, CodeBuild télécharge uniquement la branche spécifiée pour trouver la version.

Note

Pour accélérer la `DOWNLOAD_SOURCE` phase de votre build, vous pouvez également définir la profondeur de clonage de Git à un faible nombre. CodeBuild télécharge moins de versions de votre dépôt.

Rubriques

- [Spécifiez une version de GitHub référentiel avec un ID de validation](#)
- [Spécifiez une version GitHub du référentiel avec une référence et un ID de validation](#)

Spécifiez une version de GitHub référentiel avec un ID de validation

Vous pouvez spécifier une version source avec seulement un ID de validation, par exemple 12345678901234567890123467890123456789. Dans ce cas, vous CodeBuild devez télécharger l'intégralité du référentiel pour trouver la version.

Pour spécifier une version de GitHub référentiel avec un ID de validation

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#). Laissez tous les paramètres avec leurs valeurs par défaut, sauf pour les paramètres suivants.
 - Dans Source:
 - Dans le champ Source provider, sélectionnez GitHub. Si vous n'êtes pas connecté à GitHub, suivez les instructions pour vous connecter.
 - Pour Repository (Référentiel), choisissez Public repository (Référentiel public).
 - Pour Repository URL (URL du référentiel), saisissez **https://github.com/aws/aws-sdk-ruby.git**
 - Dans Environment (Environnement) :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez aws/codebuild/amazonlinux-x86_64-standard:4.0.
3. Pour Build spécification (Spécifications de génération), choisissez Insérer des commandes de génération, puis choisissez Switch to editor (Passer à l'éditeur).
4. Dans Build commands (Commandes de génération), remplacez le texte de l'espace réservé par ce qui suit :

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  build:
```

```
commands:
  - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

La section `runtime-versions` est obligatoire lorsque vous utilisez l'image standard Ubuntu 2.0. Ici, la version Ruby 2.6 de l'exécution est spécifiée, mais vous pouvez utiliser n'importe quelle exécution. La commande `echo` affiche la version du code source stocké dans la variable d'environnement `CODEBUILD_RESOLVED_SOURCE_VERSION`.

5. Dans Configuration de la génération, acceptez les valeurs par défaut, puis choisissez Démarrer la génération.
6. Pour Version de la source, saisissez **046e8b67481d53bdc86c3f6affdd5d1afae6d369**. Il s'agit du SHA d'une validation dans le référentiel <https://github.com/aws/aws-sdk-ruby.git>.
7. Choisissez Démarrer la génération.
8. Une fois la génération terminée, vous devez voir ce qui suit :
 - Sur l'onglet Journaux de génération, la version de la source du projet qui a été utilisée. Voici un exemple.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Sur l'onglet Variables d'environnement, Resolved source version (Version de source résolue) correspond à l'ID de validation utilisé pour créer la validation.
- Sur l'onglet Détail de la phase, la durée de la phase `DOWNLOAD_SOURCE`.

Spécifiez une version GitHub du référentiel avec une référence et un ID de validation

Vous pouvez spécifier une version source avec une référence et un ID de validation dans ce format : `refs/heads/branchname^{full-commit-SHA}` (par exemple, `refs/heads/main^{12345678901234567890123467890123456789}`). Dans ce cas, CodeBuild télécharge uniquement la branche spécifiée pour trouver la version.

Pour spécifier une version de GitHub référentiel avec une référence et un ID de validation.

1. Suivez les étapes de [Spécifiez une version de GitHub référentiel avec un ID de validation](#).
2. Dans le volet de navigation de gauche, choisissez Projets de génération, puis choisissez le projet que vous avez créé précédemment.
3. Choisissez Démarrer la génération.
4. Dans Version de la source, saisissez **refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}**. Il s'agit du même ID de validation et d'une référence à une branche dans le format *refs/heads/branchname^{full-commit-SHA}*.
5. Choisissez Démarrer la génération.
6. Une fois la génération terminée, vous devez voir ce qui suit :
 - Sur l'onglet Journaux de génération, la version de la source du projet qui a été utilisée. Voici un exemple.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Sur l'onglet Variables d'environnement, Resolved source version (Version de source résolue) correspond à l'ID de validation utilisé pour créer la validation.
- Sur l'onglet Détails de Détail de la phase, la durée de la phase DOWNLOAD_SOURCE doit être inférieure à la durée lorsque vous avez utilisé uniquement l'ID de validation pour spécifier la version de votre code source.

Exemples de référentiels sources tiers pour CodeBuild

Cette section décrit des exemples d'intégrations entre des référentiels sources tiers et CodeBuild

Exemple	Description
BitBucket exemple de pull request et de filtre Webhook — voir Exécutez l'exemple	Cet exemple vous montre comment créer une demande d'extraction à l'aide d'un référentiel Bitbucket. Il vous montre également comment

Exemple	Description
« Bitbucket pull request and webhook filter » pour CodeBuild	utiliser un webhook Bitbucket de manière à déclencher CodeBuild pour créer une génération de projet.
GitHub Exemple de serveur d'entreprise — voir Exécutez l'exemple de serveur d' GitHub entreprise pour CodeBuild	Cet exemple explique comment configurer vos CodeBuild projets lorsqu'un certificat est installé dans votre référentiel GitHub Enterprise Server. Il montre également comment activer les webhooks afin de CodeBuild recompiler le code source chaque fois qu'une modification de code est transmise à votre référentiel GitHub Enterprise Server.
GitHub exemple de pull request et de filtre Webhook — voir Exécutez l'exemple de GitHub pull request et de filtre webhook pour CodeBuild	Cet exemple explique comment créer une pull request à l'aide d'un référentiel GitHub Enterprise Server. Il montre également comment activer les webhooks afin de CodeBuild recompiler le code source chaque fois qu'une modification de code est transmise à votre référentiel GitHub Enterprise Server.

Exécutez l'exemple « Bitbucket pull request and webhook filter » pour CodeBuild

AWS CodeBuild prend en charge les webhooks lorsque le référentiel source est Bitbucket. Cela signifie que pour un projet de CodeBuild compilation dont le code source est stocké dans un dépôt Bitbucket, les webhooks peuvent être utilisés pour reconstruire le code source chaque fois qu'une modification de code est envoyée au référentiel. Pour de plus amples informations, veuillez consulter [Événements du webhook Bitbucket](#).

Cet exemple vous montre comment créer une demande d'extraction à l'aide d'un référentiel Bitbucket. Il vous montre également comment utiliser un webhook Bitbucket pour déclencher la création CodeBuild d'une version d'un projet.

 Note

Lors de l'utilisation de webhooks, il est possible pour un utilisateur de déclencher une compilation inattendue. Pour atténuer ce risque, voir [Bonnes pratiques d'utilisation des webhooks](#).

Rubriques

- [Conditions préalables](#)
- [Étape 1 : créer un projet de build avec Bitbucket et activer les webhooks](#)
- [Étape 2 : Déclencher un build avec un webhook Bitbucket](#)

Conditions préalables

Pour exécuter cet exemple, vous devez connecter votre AWS CodeBuild projet à votre compte Bitbucket.

 Note

CodeBuild a mis à jour ses autorisations avec Bitbucket. Si vous avez précédemment connecté votre projet à Bitbucket et que vous recevez maintenant une erreur de connexion Bitbucket, vous devez vous reconnecter pour CodeBuild autoriser la gestion de vos webhooks.

Étape 1 : créer un projet de build avec Bitbucket et activer les webhooks

Les étapes suivantes décrivent comment créer un AWS CodeBuild projet avec Bitbucket comme référentiel source et comment activer les webhooks.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Choisissez Créer un projet de génération.
4. Dans Configuration de projet :

Nom du projet

Entrez un nom pour ce projet de construction. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

5. Dans Source:

Fournisseur de source

Choisissez Bitbucket. Suivez les instructions pour vous connecter (ou vous reconnecter) à Bitbucket, puis choisissez Autoriser.

Référentiel

Choisissez Repository dans mon compte Bitbucket.

Si vous n'êtes pas encore connecté à votre compte Bitbucket, entrez votre nom d'utilisateur Bitbucket et le mot de passe de l'application, puis sélectionnez Enregistrer les informations d'identification Bitbucket.

Référentiel Bitbucket

Entrez l'URL de votre dépôt Bitbucket.

6. Dans Événements webhook de la source principale, sélectionnez ce qui suit.

Note

La section des événements du webhook de la source principale n'est visible que si vous avez choisi Repository in my Bitbucket account à l'étape précédente.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.

4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations sur les types d'événements et les filtres Bitbucket Webhook, consultez.

[Événements du webhook Bitbucket](#)

7. Dans Environment (Environnement) :

Image de l'environnement

Sélectionnez l'une des méthodes suivantes :

Pour utiliser une image Docker gérée par AWS CodeBuild :

Choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.

Pour utiliser une autre image Docker :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS

Pour utiliser une image Docker privée :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur.

Rôle de service

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

 Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

8. Dans Buildspec, effectuez l'une des opérations suivantes :

- Choisissez Utiliser un fichier buildspec pour utiliser le fichier buildspec.yml dans le répertoire racine du code source.
- Choisissez Insérer des commandes de génération pour utiliser la console pour insérer des commandes de génération.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

9. Dans Artefacts:

Type

Sélectionnez l'une des méthodes suivantes :

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.

- Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
- Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, `appspec.yml`, `target/my-app.jar`). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

Configuration supplémentaire

Développez Additional configuration (Configuration supplémentaire) et définissez les options dont vous avez besoin.

10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Étape 2 : Déclencher un build avec un webhook Bitbucket

Pour un projet qui utilise des webhooks Bitbucket, AWS CodeBuild crée une version lorsque le référentiel Bitbucket détecte une modification dans votre code source.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération, puis sélectionnez un projet associés à un référentiel Bitbucket avec webhooks. Pour plus d'informations sur la création d'un projet Webhook Bitbucket, consultez [the section called "Étape 1 : créer un projet de build avec Bitbucket et activer les webhooks"](#).
3. Modifiez le code du référentiel Bitbucket de votre projet.
4. Créez une demande d'extraction sur votre référentiel Bitbucket. Pour de plus amples informations, veuillez consulter des informations sur la [création d'une demande d'extraction](#).
5. Dans la page des webhooks Bitbucket, choisissez View request (Afficher la demande) pour afficher la liste des événements récents.
6. Choisissez Afficher les détails pour voir les détails de la réponse renvoyée par CodeBuild. Elle peut ressembler à ceci :

```
"response":"Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
```

```
"statusCode":200
```

7. Accédez à la page de demande d'extraction Bitbucket pour voir le statut de la génération.

Exécutez l'exemple de serveur d' GitHub entreprise pour CodeBuild

AWS CodeBuild prend en charge GitHub Enterprise Server en tant que référentiel source. Cet exemple montre comment configurer vos CodeBuild projets lorsqu'un certificat est installé dans votre référentiel GitHub Enterprise Server. Il montre également comment activer les webhooks afin de CodeBuild recompiler le code source chaque fois qu'une modification de code est transmise à votre référentiel GitHub Enterprise Server.

Rubriques

- [Conditions préalables](#)
- [Étape 1 : créer un projet de build avec GitHub Enterprise Server et activer les webhooks](#)

Conditions préalables

1. Générez un jeton d'accès personnel pour votre CodeBuild projet. Nous vous recommandons de créer un utilisateur GitHub Enterprise et de générer un jeton d'accès personnel pour cet utilisateur. Copiez-le dans votre presse-papiers afin de pouvoir l'utiliser lors de la création de votre CodeBuild projet. Pour plus d'informations, consultez la section [Création d'un jeton d'accès personnel pour la ligne de commande](#) sur le site Web d' GitHub aide.

Lorsque vous créez le jeton d'accès personnel, incluez la portée du référentiel dans la définition.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

☒ repo	Full control of private repositories
☒ repo:status	Access commit status
☒ repo_deployment	Access deployment status
☒ public_repo	Access public repositories

2. Téléchargez votre certificat depuis GitHub Enterprise Server. CodeBuild utilise le certificat pour établir une connexion SSL fiable avec le référentiel.

Clients Linux/macOS :

À partir d'une fenêtre de terminal , exécutez la commande suivante :

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```

Remplacez les espaces réservés de la commande par les valeurs suivantes :

HOST. Adresse IP de votre référentiel GitHub Enterprise Server.

PORTNUMBER. Le numéro de port que vous utilisez pour vous connecter (par exemple, 443).

folder. Le dossier dans lequel vous avez téléchargé votre certificat.

filename. Le nom de fichier de votre fichier de certificat.

 Important

Enregistrez le certificat en tant que fichier .pem.

Clients Windows:

Utilisez votre navigateur pour télécharger votre certificat depuis GitHub Enterprise Server. Pour afficher les détails de certificat du site, choisissez l'icône en forme de cadenas. Pour plus d'informations sur la manière d'exporter le certificat, consultez la documentation de votre navigateur.

 Important

Enregistrez le certificat en tant que fichier .pem.

3. Chargez votre fichier de certificat dans un compartiment S3. Pour de plus amples informations sur la façon de créer un compartiment S3, veuillez consulter [Comment créer un compartiment S3 ?](#) Pour de plus amples informations sur le chargement d'objets dans un compartiment S3, veuillez consulter [Comment charger des fichiers ou dossiers vers un compartiment ?](#)

 Note

Ce compartiment doit se trouver dans la même AWS région que vos builds. Par exemple, si vous demandez d' CodeBuild exécuter un build dans la région USA Est (Ohio), le bucket doit se trouver dans la région USA Est (Ohio).

Étape 1 : créer un projet de build avec GitHub Enterprise Server et activer les webhooks

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, dans Source provider, choisissez GitHub Enterprise Server.
 - Choisissez Gérer les informations d'identification du compte, puis sélectionnez Jeton d'accès personnel. Pour Service, choisissez Secrets Manager (recommandé) et configurez votre secret. Ensuite, dans Jeton d'accès personnel GitHub Enterprise, entrez votre jeton d'accès personnel et choisissez Enregistrer.
 - Dans URL du référentiel, saisissez le chemin d'accès à votre référentiel, y compris le nom du référentiel.
 - Développez Additional configuration (Configuration supplémentaire).
 - Sélectionnez Rebuild every time a code change is pushed to this repository (Regénérer chaque fois qu'une modification de code est transmise à ce référentiel) pour régénérer chaque fois qu'une modification de code est transmise à ce référentiel.
 - Sélectionnez Activer le protocole SSL non sécurisé pour ignorer les avertissements SSL lorsque vous vous connectez au référentiel de votre projet GitHub Enterprise Server.

Note

Nous vous recommandons d'utiliser Insecure SSL (SSL non sécurisé) uniquement pour les tests. Cette option ne doit pas être utilisée dans un environnement de production.

Source

Add source

Source 1 - Primary

Source provider

GitHub Enterprise▼

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ Additional configuration

Git clone depth, Insecure SSL

Git clone depth - optional

1▼

Webhook - optional

☒ Rebuild every time a code change is pushed to this repository

Branch filter - optional

Enter a regular expression

Insecure SSL - optional

Enable this flag to ignore SSL warnings while connecting to project source.

☐ Enable insecure SSL

5. Dans Environment (Environnement) :

Pour Image d'environnement, effectuez l'une des actions suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

6. Dans Rôle du service, sélectionnez l'une des options suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

 Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

7. Développez Additional configuration (Configuration supplémentaire).

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources du VPCs

Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

8. Dans Buildspec, effectuez l'une des opérations suivantes :

- Choisissez Utiliser un fichier buildspec pour utiliser le fichier buildspec.yml dans le répertoire racine du code source.
- Choisissez Insérer des commandes de génération pour utiliser la console pour insérer des commandes de génération.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

9. Dans Artefacts, pour Type, effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

10. Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :

- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

 Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

 Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Des mises en cache pour améliorer les performances](#).

11. Choisissez Créer un projet de génération. Sur la page de projet de génération, choisissez Démarrer la génération.

Exécutez l'exemple de GitHub pull request et de filtre webhook pour CodeBuild

AWS CodeBuild prend en charge les webhooks lorsque le référentiel source l'est GitHub. Cela signifie que pour un projet de CodeBuild compilation dont le code source est stocké dans un GitHub référentiel, les webhooks peuvent être utilisés pour reconstruire le code source chaque fois qu'une modification de code est envoyée au référentiel. Pour CodeBuild des exemples, voir [AWS CodeBuild Échantillons](#).

Note

Lors de l'utilisation de webhooks, il est possible pour un utilisateur de déclencher une compilation inattendue. Pour atténuer ce risque, voir [Bonnes pratiques d'utilisation des webhooks](#).

Rubriques

- [Étape 1 : créer un projet de construction avec GitHub et activer les webhooks](#)
- [Étape 2 : Vérifiez que les webhooks sont activés](#)

Étape 1 : créer un projet de construction avec GitHub et activer les webhooks

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Choisissez Créer un projet de génération.
4. Dans Configuration de projet :

Nom du projet

Entrez un nom pour ce projet de construction. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

5. Dans Source:

Fournisseur de source

Sélectionnez GitHub. Suivez les instructions pour vous connecter (ou vous reconnecter) à, GitHub puis choisissez Autoriser.

Référentiel

Choisissez Repository dans mon GitHub compte.

GitHub référentiel

Entrez l'URL de votre GitHub dépôt.

6. Dans Événements webhook de la source principale, sélectionnez ce qui suit.

Note

La section des événements du webhook de la source principale n'est visible que si vous avez choisi Repository dans mon GitHub compte à l'étape précédente.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations sur les types d'événements et les filtres GitHub Webhook, consultez [GitHub événements webhook](#).

7. Dans Environment (Environnement) :

Image de l'environnement

Sélectionnez l'une des méthodes suivantes :

Pour utiliser une image Docker gérée par AWS CodeBuild :

Choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.

Pour utiliser une autre image Docker :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS

Pour utiliser une image Docker privée :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur.

Rôle de service

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour

associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

8. Dans Buildspec, effectuez l'une des opérations suivantes :

- Choisissez Utiliser un fichier buildspec pour utiliser le fichier buildspec.yml dans le répertoire racine du code source.
- Choisissez Insérer des commandes de génération pour utiliser la console pour insérer des commandes de génération.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

9. Dans Artefacts:

Type

Sélectionnez l'une des méthodes suivantes :

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

Configuration supplémentaire

Développez Additional configuration (Configuration supplémentaire) et définissez les options dont vous avez besoin.

10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Étape 2 : Vérifiez que les webhooks sont activés

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :
 - Cliquez sur le lien du projet de build avec des webhooks que vous souhaitez réviser, puis choisissez Build details (Détails de build).
 - Cliquez sur le bouton à côté du projet de génération avec les webhooks que vous souhaitez vérifier, choisissez Afficher les détails, puis choisissez l'onglet Détails de la construction.
4. Dans Événements du webhook de la source principale, choisissez le lien URL du webhook.
5. Dans votre GitHub dépôt, sur la page Paramètres, sous Webhooks, vérifiez que les Pull Requests et Push sont sélectionnés.
6. Dans les paramètres de votre GitHub profil, sous Paramètres personnels, Applications, OAuthApplications autorisées, vous devriez voir que votre application a été autorisée à accéder à la AWS région que vous avez sélectionnée.

Tutoriel : signature de code Apple avec Fastlane lors de CodeBuild l'utilisation de S3 pour le stockage de certificats

[fastlane](#) est un outil d'automatisation open source populaire qui permet d'automatiser les déploiements et les versions bêta de vos applications iOS et Android. Il gère toutes les tâches fastidieuses, comme la génération de captures d'écran, la signature de code et la publication de votre application.

Conditions préalables

Pour terminer ce didacticiel, vous devez d'abord avoir configuré les éléments suivants :

- Un Compte AWS
- Un [compte Apple Developer](#)
- Un compartiment S3 pour le stockage des certificats
- fastlane installé dans votre projet - [Guide](#) d'installation de fastlane

Étape 1 : configurer Fastlane Match avec S3 sur votre machine locale

[Fastlane Match](#) est l'un des [outils de Fastlane](#). Il permet une configuration fluide pour la signature de code, à la fois dans votre environnement de développement local et ailleurs. CodeBuild Fastlane Match stocke tous vos certificats de signature de code et profils de provisionnement dans un stockage Git repository/S3 Bucket/Google Cloud, et télécharge et installe les certificats et profils nécessaires en cas de besoin.

Dans cet exemple de configuration, vous allez configurer et utiliser un compartiment Amazon S3 pour le stockage.

1. Initialisez match dans votre projet :

```
fastlane match init
```

2. Lorsque vous y êtes invité, choisissez S3 comme mode de stockage.
3. Mettez à jour votre « Matchfile » pour utiliser S3 :

```
storage_mode("s3")
s3_bucket("your-s3-bucket-name")
s3_region("your-aws-region")
type("appstore") # The default type, can be: appstore, adhoc, enterprise or
development
```

Étape 2 : Configurez votre Fastfile

Créez ou mettez à jour votre `Fastfile` avec la ligne suivante.

Activé CodeBuild, Fastlane Match devra être exécuté chaque fois que vous créez et signez votre application. Le moyen le plus simple de le faire est d'ajouter l'`match` action à la voie qui crée votre application.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

Note

Assurez-vous d'ajouter des `setup_ci` éléments à la `before_all` section Fastfile pour que l'action de `match` fonctionne correctement. Cela garantit qu'un trousseau Fastlane temporaire doté des autorisations appropriées est utilisé. Si vous ne l'utilisez pas, vous risquez de rencontrer des échecs de compilation ou des résultats incohérents.

Étape 3 : Exécutez la **fastlane match** commande pour générer les certificats et profils respectifs

La commande `fastlane match` pour le type donné (par exemple, `development`, `appstore`, `adhoc`, `enterprise`) générera le certificat et le profil s'ils ne sont pas disponibles dans le magasin distant. Les certificats et les profils seront stockés dans S3 par `fastlane`.

```
bundle exec fastlane match appstore
```

L'exécution de la commande sera interactive et `fastlane` demandera de définir une phrase secrète pour déchiffrer les certificats.

Étape 4 : Créez le dossier de candidature pour votre projet

Créez ou ajoutez le fichier de candidature en fonction de votre projet.

1. Créez ou ajoutez le [Gymfile, Appfile, Snapfile, Deliverfile](#) en fonction des exigences de construction de votre projet.
2. Validez les modifications dans votre dépôt distant

Étape 5 : Création de variables d'environnement dans Secrets Manager

Créez deux secrets pour stocker le cookie de session Fastlane et le mot de passe correspondant. Pour plus d'informations sur la création de secrets dans Secrets Manager, voir [Création d'un AWS Secrets Manager secret](#).

1. Accédez à votre cookie de session Fastlane comme suit.
 - a. Clé secrète - FASTLANE_SESSION
 - b. Valeur secrète : cookie de session généré par l'exécution de la commande suivante sur votre ordinateur local.

Note

Cette valeur est disponible après authentification dans un fichier local : `~/fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <apple account>
```

2. Phrase de passe Fastlane Match - Pour permettre à Fastlane Match de déchiffrer les certificats et les profils stockés dans le compartiment S3, il est nécessaire d'ajouter le mot de passe de chiffrement que vous avez configuré à l'étape de configuration de Match aux variables d'environnement du projet. CodeBuild
 - a. Clé secrète - MATCH_PASSWORD
 - b. Valeur secrète - *<match passphrase to decrypt certificates>*. Le mot de passe est défini lors de la génération des certificats à l'étape 3.

 Note

Lorsque vous créez les secrets ci-dessus dans Secrets Manager, n'oubliez pas de donner un nom au secret avec le préfixe suivant : /CodeBuild/

Étape 6 : Création d'un parc informatique

Créez le parc informatique pour votre projet.

1. Dans la console, accédez à CodeBuild et créez un nouveau parc informatique.
2. Choisissez « macOS » comme système d'exploitation et sélectionnez le type de calcul et l'image appropriés.

Étape 7 : Création d'un projet dans CodeBuild

Créez votre projet dans CodeBuild.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
3. Configurez votre fournisseur de source (tel que GitHub, CodeCommit). Il s'agit du référentiel des sources du projet iOS et non du référentiel de certificats.
4. Dans Environment (Environnement) :
 - Choisissez Capacité réservée.
 - Pour Fleet, sélectionnez la flotte créée ci-dessus.
 - Indiquez le nom du rôle de service qui CodeBuild sera créé pour vous.
 - Fournissez les variables d'environnement ci-dessous.
 - Nom :MATCH_PASSWORD, Valeur :<secrets arn>, Type : Secrets Manager (Secrets ARN créé à l'étape 5 pour MATCH_PASSWORD)
 - Nom :FASTLANE_SESSION, Valeur :<secrets arn>, Type : Secrets Manager (ARN de secrets créé à l'étape 5 pour FASTLANE_SESSION)

5. Dans Buildspec, ajoutez ce qui suit :

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Étape 8 : Configuration du rôle IAM

Une fois le projet créé, assurez-vous que le rôle de service de votre CodeBuild projet est autorisé à accéder au compartiment S3 contenant les certificats. Ajoutez la politique suivante au rôle :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::your-s3-bucket-name"
    },
    {
```

```
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:DeleteObject"
        ],
        "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
    }
}
```

Étape 9 : Exécuter le build

Exécutez la génération. Vous pouvez consulter l'état de la construction et vous connecter CodeBuild.

Une fois la tâche terminée, vous pourrez consulter le journal de la tâche.

Résolution des problèmes

- Si vous rencontrez des problèmes lors de l'extraction des certificats, assurez-vous que vos autorisations IAM sont correctement configurées pour l'accès S3.
- Si vous rencontrez des problèmes lors du déchiffrement des certificats, assurez-vous de définir le mot de passe correct dans la variable d'environnement MATCH_PASSWORD.
- Pour les problèmes de signature de code, vérifiez que votre compte Apple Developer possède les certificats et profils nécessaires, et que l'identifiant du bundle de votre projet Xcode correspond à celui de votre profil d'approvisionnement.

Considérations sur la sécurité

Les considérations de sécurité suivantes concernent ce didacticiel.

- Assurez-vous que votre compartiment S3 possède les paramètres de sécurité appropriés, y compris le chiffrement au repos. Assurez-vous en particulier que le bucket n'est pas accessible au public et limitez l'accès uniquement CodeBuild au système qui a besoin d'un accès.
- Envisagez de l'utiliser AWS Secrets Manager pour stocker des informations sensibles telles que MATCH_PASSWORD et FASTLANE_SESSION.

Cet exemple fournit une configuration pour la signature de code iOS avec Fastlane lors de CodeBuild l'utilisation d'Amazon S3 pour le stockage de certificats. Vous devrez peut-être ajuster certaines étapes en fonction des exigences et de CodeBuild l'environnement spécifiques de votre projet. Cette approche tire parti des AWS services pour améliorer la sécurité et l'intégration au sein de l' AWS écosystème.

Tutoriel : signature de code Apple avec Fastlane lors de CodeBuild l'utilisation GitHub pour le stockage de certificats

[fastlane](#) est un outil d'automatisation open source populaire qui permet d'automatiser les déploiements et les versions bêta de vos applications iOS et Android. Il gère toutes les tâches fastidieuses, comme la génération de captures d'écran, la signature de code et la publication de votre application.

Cet exemple montre comment configurer la signature de code Apple à l'aide de Fastlane dans un CodeBuild projet exécuté sur un parc Mac, en GitHub tant que stockage pour les certificats et les profils de provisionnement.

Conditions préalables

Pour terminer ce didacticiel, vous devez d'abord avoir configuré les éléments suivants :

- Un Compte AWS
- Un [compte Apple Developer](#)
- Un GitHub référentiel privé pour le stockage des certificats
- fastlane installé dans votre projet - [Guide](#) d'installation de fastlane

Étape 1 : Configurez Fastlane Match avec GitHub sur votre machine locale

[Fastlane Match](#) est l'un des [outils de Fastlane](#). Il permet une configuration fluide pour la signature de code, à la fois dans votre environnement de développement local et ailleurs. CodeBuild Fastlane Match stocke tous vos certificats de signature de code et profils de provisionnement dans un stockage Git repository/S3 Bucket/Google Cloud, et télécharge et installe les certificats et profils nécessaires en cas de besoin.

Dans cet exemple de configuration, nous allons configurer et utiliser un dépôt Git pour le stockage.

1. Initialisez match dans votre projet :

```
fastlane match init
```

2. Lorsque vous y êtes invité, choisissez GitHub le mode de stockage.
3. Mettez à jour votre « Matchfile » pour utiliser GitHub :

```
git_url("https://github.com/your-username/your-certificate-repo.git")
storage_mode("git")
type("development") # The default type, can be: appstore, adhoc, enterprise or
development
```

Note

Assurez-vous de saisir l'URL HTTPS de votre dépôt Git pour que fastlane puisse s'authentifier et cloner avec succès. Dans le cas contraire, une erreur d'authentification peut s'afficher lorsque vous tentez d'utiliser Match.

Étape 2 : Configurez votre Fastfile

Créez ou mettez à jour votre `Fastfile` avec la ligne suivante.

Activé CodeBuild, Fastlane Match devra être exécuté chaque fois que vous créez et signez votre application. Le moyen le plus simple de le faire est d'ajouter l'`match` action à la voie qui crée votre application.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

```
end  
end
```

Note

Assurez-vous d'ajouter des `setup_ci` éléments à la `before_all` section `Fastfile` pour que l'action de `match` fonctionne correctement. Cela garantit qu'un trousseau Fastlane temporaire doté des autorisations appropriées est utilisé. Si vous ne l'utilisez pas, vous risquez de rencontrer des échecs de compilation ou des résultats incohérents.

Étape 3 : Exécutez la **fastlane match** commande pour générer les certificats et profils respectifs

La commande `fastlane match` pour le type donné (c'est-à-dire `development`, `appstore`, `adhoc`, `enterprise`) générera le certificat et le profil s'ils ne sont pas disponibles dans le magasin distant. Les certificats et les profils seront stockés GitHub par Fastlane.

```
bundle exec fastlane match appstore
```

L'exécution de la commande sera interactive et `fastlane` demandera de définir une phrase secrète pour déchiffrer les certificats.

Étape 4 : Créez le dossier de candidature pour votre projet

Créez ou ajoutez le fichier de candidature en fonction de votre projet.

1. Créez ou ajoutez le [Gymfile](#), [Appfile](#), [Snapfile](#), [Deliverfile](#) en fonction des exigences de construction de votre projet.
2. Validez les modifications dans votre dépôt distant.

Étape 5 : Création de variables d'environnement dans Secrets Manager

Créez trois secrets pour stocker le cookie de session Fastlane et le mot de passe correspondant. Pour plus d'informations sur la création de secrets dans Secrets Manager, voir [Création d'un AWS Secrets Manager secret](#).

1. Accédez à votre cookie de session Fastlane comme suit.
 - a. Clé secrète - FASTLANE_SESSION
 - b. Valeur secrète : cookie de session généré par l'exécution de la commande suivante sur votre ordinateur local.

 Note

Cette valeur est disponible après authentification dans un fichier local : `~/.fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <Apple_account>
```

2. Phrase de passe Fastlane Match - Pour permettre à Fastlane Match de déchiffrer les certificats et les profils stockés dans le référentiel Git, il est nécessaire d'ajouter le mot de passe de chiffrement que vous avez configuré à l'étape de configuration de Match aux variables d'environnement du projet. CodeBuild
 - a. Clé secrète - MATCH_PASSWORD
 - b. Valeur secrète - `<match passphrase to decrypt certificates>`. Le mot de passe est défini lors de la génération des certificats à l'étape 3.
3. Fastlane *MATCH_GIT_BASIC_AUTHORIZATION* - définissez une autorisation de base pour le match :
 - a. Clé secrète :

`MATCH_GIT_BASIC_AUTHORIZATION`
 - b. Valeur secrète - La valeur doit être une chaîne codée en base64 de votre nom d'utilisateur et de votre jeton d'accès personnel (PAT) au format `username:password`. Vous pouvez le générer à l'aide de la commande suivante :

```
echo -n your_github_username:your_personal_access_token | base64
```

Vous pouvez générer votre PAT sur la GitHub console dans Votre profil > Paramètres > Paramètres des développeurs > Jeton d'accès personnel. Pour plus d'informations,

consultez le guide suivant : <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>.

Note

Lorsque vous créez les secrets ci-dessus dans Secrets Manager, n'oubliez pas de leur donner un nom avec le préfixe suivant : /CodeBuild/

Étape 6 : Création d'un parc informatique

Créez le parc informatique pour votre projet.

1. Dans la console, accédez à CodeBuild et créez un nouveau parc informatique.
2. Choisissez macOS comme système d'exploitation et sélectionnez le type de calcul et l'image appropriés.

Étape 7 : Création d'un projet dans CodeBuild

Créez votre projet dans CodeBuild.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
3. Configurez votre fournisseur de source (tel que GitHub, CodeCommit). Il s'agit du référentiel des sources du projet iOS et non du référentiel de certificats.
4. Dans Environment (Environnement) :
 - Choisissez Capacité réservée.
 - Pour Fleet, sélectionnez la flotte créée ci-dessus.
 - Indiquez le nom du rôle de service qui CodeBuild sera créé pour vous.
 - Fournissez les variables d'environnement ci-dessous.
 - Nom :MATCH_PASSWORD, Valeur :<secrets arn>, Type : Secrets Manager (Secrets ARN créé à l'étape 5 pour MATCH_PASSWORD)

- Nom :FASTLANE_SESSION, Valeur :<secrets arn>, Type : Secrets Manager (ARN de secrets créé à l'étape 5 pour FASTLANE_SESSION)
- Nom :MATCH_GIT_BASIC_AUTHORIZATION, Valeur :<secrets ARN>, Type : Secrets Manager Secrets ARN (créé à l'étape 5 pourMATCH_GIT_BASIC_AUTHORIZATION)

5. Dans Buildspec, ajoutez ce qui suit :

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Étape 8 : Exécuter le build

Exécutez la génération. Vous pouvez consulter l'état de la construction et vous connecter CodeBuild.

Une fois la tâche terminée, vous pourrez consulter le journal de la tâche.

Résolution des problèmes

- Si vous rencontrez des problèmes pour accéder au GitHub dépôt, vérifiez votre jeton d'accès personnel et la variable d'environnement MATCH_GIT_BASIC_AUTHORIZATION.
- Si vous rencontrez des problèmes lors du déchiffrement des certificats, assurez-vous de définir le mot de passe correct dans la variable d'environnement MATCH_PASSWORD.

- Pour les problèmes de signature de code, vérifiez que votre compte Apple Developer possède les certificats et profils nécessaires et que l'identifiant du bundle de votre projet Xcode correspond à celui de votre profil d'approvisionnement.

Considérations sur la sécurité

Les considérations de sécurité suivantes concernent ce didacticiel.

- Préservez la GitHub confidentialité de votre référentiel de certificats et contrôlez régulièrement l'accès.
- Envisagez de l'utiliser AWS Secrets Manager pour stocker des informations sensibles telles que `MATCH_PASSWORD` et `FASTLANE_SESSION`.

Cet exemple fournit une configuration pour la signature de code iOS avec Fastlane lors de CodeBuild l'utilisation GitHub pour le stockage de certificats. Vous devrez peut-être ajuster certaines étapes en fonction des exigences et de CodeBuild l'environnement spécifiques de votre projet. Cette approche tire parti des AWS services pour améliorer la sécurité et l'intégration au sein de l' AWS écosystème.

Définissez les noms des artefacts au moment de la construction à l'aide du versionnement sémantique

Cet exemple contient des exemples de fichiers buildspec qui montrent comment spécifier le nom d'un artefact qui est créé au moment de la génération. Un nom spécifié dans un fichier buildspec peut comprendre des commandes Shell et des variables d'environnement afin de le rendre unique. Un nom que vous spécifiez dans un fichier buildspec se substitue au nom que vous saisissez dans la console lorsque vous créez votre projet.

Si vous procédez à plusieurs générations, le nom d'artefact spécifié dans le fichier buildspec permet de vous assurer que les noms de vos fichiers d'artefact de sortie sont uniques. Par exemple, vous pouvez utiliser une date et un horodatage inséré dans le nom d'un artefact au moment de la génération.

Si vous souhaitez remplacer le nom de l'artefact que vous avez saisi dans la console par un nom dans le fichier buildspec, procédez comme suit :

1. Configurez votre projet de génération afin de remplacer le nom de l'artefact par un nom du fichier buildspec.

- Si vous utilisez la console pour créer votre projet de génération, sélectionnez Activer la gestion sémantique des versions. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).
 - Si vous utilisez le AWS CLI, définissez le sur `overrideArtifactName` true dans le fichier au format JSON transmis à `create-project` Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).
 - Si vous utilisez l' AWS CodeBuild API, placez l'`overrideArtifactName`indicateur sur l'`ProjectArtifact`sobjet lorsqu'un projet est créé ou mis à jour ou qu'une construction est lancée.
2. Spécifiez un nom dans le fichier `buildspec`. Utilisez l'exemple de fichier `buildspec` suivant comme guide.

Cet exemple Linux vous montre comment spécifier un nom d'artefact qui comprend la date de création de la génération :

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${date +%Y-%m-%d}
```

Cet exemple Linux vous montre comment spécifier un nom d'artefact qui utilise une variable d'environnement CodeBuild. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

Cet exemple Windows vous montre comment spécifier un nom d'artefact qui comprend la date et l'heure de création de la génération :

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Cet exemple Windows montre comment spécifier un nom d'artefact qui utilise une variable déclarée dans le fichier buildspec et une variable d'environnement. CodeBuild Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Pour de plus amples informations, veuillez consulter [Référence de spécification de construction pour CodeBuild](#).

Exécutez des exemples Microsoft Windows pour CodeBuild

Ces exemples utilisent un environnement de AWS CodeBuild génération exécutant Microsoft Windows Server 2019, le .NET Framework et le SDK .NET Core pour créer des fichiers d'exécution à partir de code écrit en F# et Visual Basic.

Important

L'analyse de ces échantillons peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [CodeBuild sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatch Tarification Amazon](#).

Exécutez les exemples Windows

Pour exécuter les exemples Windows, procédez comme suit.

Pour exécuter les exemples Windows

1. Créez les fichiers comme décrit dans les [Fichiers](#) sections [Structure de répertoire](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 CodeCommit ou un GitHub référentiel or.

Important

Ne chargez pas *(root directory name)*, mais seulement les fichiers à l'intérieur de *(root directory name)*.

Si vous utilisez un compartiment d'entrée S3, veillez à créer un fichier ZIP contenant les fichiers, puis chargez ce fichier dans le compartiment d'entrée. N'ajoutez pas *(root directory name)* au fichier ZIP, mais seulement les fichiers à l'intérieur de *(root directory name)*.

2. Créez un projet de génération. Le projet de génération doit utiliser `l'mcr.microsoft.com/dotnet/framework/sdk:4.8` image pour créer des projets .NET Framework.

Si vous utilisez le AWS CLI pour créer le projet de construction, l'entrée au format JSON de la `create-project` commande peut ressembler à ceci. (Remplacez les espaces réservés par vos propres valeurs.)

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Exécutez le build et suivez les étapes décrites dans [Exécuter les builds manuellement](#).
4. Pour obtenir l'artefact de sortie de génération, dans le compartiment de sortie S3, téléchargez le fichier `windows-build-output-artifact.zip` sur votre ordinateur ou instance local. Extrayez le contenu pour accéder au runtime et aux autres fichiers.
 - Le fichier d'exécution de l'exemple F# utilisant le .NET Framework se trouve dans le `FSharpHelloWorld\bin\Debug` répertoire. `FSharpHelloWorld.exe`
 - Le fichier d'exécution de l'exemple Visual Basic utilisant le .NET Framework se trouve dans le `VBHelloWorld\bin\Debug` répertoire. `VBHelloWorld.exe`

Structure de répertoire

Ces exemples supposent les structures de répertoire suivantes.

F# et .NET Framework

```
(root directory name)
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
### App.config
### AssemblyInfo.fs
### FSharpHelloWorld.fsproj
### Program.fs
```

Visual Basic et .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
### App.config
### HelloWorld.vb
### VBHelloWorld.vbproj
### My Project
### Application.Designer.vb
### Application.myapp
### AssemblyInfo.vb
### Resources.Designer.vb
### Resources.resx
### Settings.Designer.vb
### Settings.settings
```

Fichiers

Ces exemples utilisent les fichiers suivants.

F# et .NET Framework

buildspec.yml (dans *(root directory name)*):

```
version: 0.2

env:
```

```

variables:
  SOLUTION: .\FSharpHelloWorld.sln
  PACKAGE_DIRECTORY: .\packages
  DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln (dans *(root directory name)*):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config (dans *(root directory name)\FSharpHelloWorld*):

```
<?xml version="1.0" encoding="utf-8" ?>
```



```
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

AssemblyInfo.fs (dans *(root directory name)\FSharpHelloWorld*):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
```

```
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]

do
  ()
```

FSharpHelloWorld.fsproj (dans *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <Tailcalls>>false</Tailcalls>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
```

```

    <Optimize>true</Optimize>
    <Tailcalls>true</Tailcalls>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="mscorlib" />
    <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
      <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Numerics" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="AssemblyInfo.fs" />
    <Compile Include="Program.fs" />
    <None Include="App.config" />
  </ItemGroup>
  <PropertyGroup>
    <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''">11</
MinimumVisualStudioVersion>
  </PropertyGroup>
  <Choose>
    <When Condition="'$(VisualStudioVersion)' == '11.0'">
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </When>
    <Otherwise>
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </Otherwise>
  </Choose>

```

```

<Import Project="$(FSharpTargetsPath)" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Program.fs (dans *(root directory name)*\FSharpHelloWorld):

```

// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code

```

Visual Basic et .NET Framework

buildspec.yml (dans *(root directory name)*):

```

version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
        PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
        p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
        \Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
    artifacts:
      files:

```

```
- .\VBHelloWorld\bin\Debug\*
```

VBHelloWorld.sln (dans *(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal
```

App.config (dans *(root directory name)\VBHelloWorld*):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
    </startup>
</configuration>
```

HelloWorld.vb (dans *(root directory name)\VBHelloWorld*):

```
Module HelloWorld

    Sub Main()
        MsgBox("Hello World")
    End Sub
End Module
```

End Sub

End Module

VBHelloWorld.vbproj (dans (*root directory name*)\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
```

```

    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
</PropertyGroup>
<PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
</PropertyGroup>
<PropertyGroup>
    <OptionCompare>Binary</OptionCompare>
</PropertyGroup>
<PropertyGroup>
    <OptionStrict>Off</OptionStrict>
</PropertyGroup>
<PropertyGroup>
    <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <Reference Include="System.Xml" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
    <Import Include="Microsoft.VisualBasic" />
    <Import Include="System" />
    <Import Include="System.Collections" />
    <Import Include="System.Collections.Generic" />
    <Import Include="System.Data" />
    <Import Include="System.Diagnostics" />
    <Import Include="System.Linq" />
    <Import Include="System.Xml.Linq" />
    <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
    <Compile Include="HelloWorld.vb" />
    <Compile Include="My Project\AssemblyInfo.vb" />
    <Compile Include="My Project\Application.Designer.vb">
        <AutoGen>True</AutoGen>
        <DependentUpon>Application.myapp</DependentUpon>
    </Compile>
    <Compile Include="My Project\Resources.Designer.vb">

```

```

    <AutoGen>True</AutoGen>
    <DesignTime>True</DesignTime>
    <DependentUpon>Resources.resx</DependentUpon>
  </Compile>
  <Compile Include="My Project\Settings.Designer.vb">
    <AutoGen>True</AutoGen>
    <DependentUpon>Settings.settings</DependentUpon>
    <DesignTimeSharedInput>True</DesignTimeSharedInput>
  </Compile>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="My Project\Resources.resx">
    <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.vb</LastGenOutput>
    <CustomToolNamespace>My.Resources</CustomToolNamespace>
    <SubType>Designer</SubType>
  </EmbeddedResource>
</ItemGroup>
<ItemGroup>
  <None Include="My Project\Application.myapp">
    <Generator>MyApplicationCodeGenerator</Generator>
    <LastGenOutput>Application.Designer.vb</LastGenOutput>
  </None>
  <None Include="My Project\Settings.settings">
    <Generator>SettingsSingleFileGenerator</Generator>
    <CustomToolNamespace>My</CustomToolNamespace>
    <LastGenOutput>Settings.Designer.vb</LastGenOutput>
  </None>
  <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.

    Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
-->
</Project>

```

Application.Designer.vb (dans *(root directory name)*\VBHelloWorld\My Project):


```
' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----
```

```
Option Strict On
Option Explicit On
```

Application.myapp (dans *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb (dans *(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
```

```

<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>

'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>

```

Resources.Designer.vb (dans *(root directory name)\VBHelloWorld\My Project*):

```

'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On

Namespace My.Resources

    'This class was auto-generated by the StronglyTypedResourceBuilder
    'class via a tool like ResGen or Visual Studio.
    'To add or remove a member, edit your .ResX file then rerun ResGen

```

```

'with the /str option, or rebuild your VS project.
'''<summary>
'''  A strongly-typed resource class, for looking up localized strings, etc.
'''</summary>

<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
Friend Module Resources

    Private resourceMan As Global.System.Resources.ResourceManager

    Private resourceCulture As Global.System.Globalization.CultureInfo

    '''<summary>
    '''  Returns the cached ResourceManager instance used by this class.
    '''</summary>

    <Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
    -
    Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
        Get
            If Object.ReferenceEquals(resourceMan, Nothing) Then
                Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
                resourceMan = temp
            End If
            Return resourceMan
        End Get
    End Property

    '''<summary>
    '''  Overrides the current thread's CurrentUICulture property for all
    '''  resource lookups using this strongly typed resource class.
    '''</summary>

    <Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
    -
    Friend Property Culture() As Global.System.Globalization.CultureInfo
        Get

```

```

        Return resourceCulture
    End Get
    Set(ByVal value As Global.System.Globalization.CultureInfo)
        resourceCulture = value
    End Set
End Property
End Module
End Namespace

```

Resources.resx (dans (*root directory name*)\VBHelloWorld\My Project):

```

<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema

    Version 2.0

    The primary goals of this format is to allow a simple XML format
    that is mostly human readable. The generation and parsing of the
    various data types are done through the TypeConverter classes
    associated with the data types.

    Example:

    ... ado.net/XML headers & schema ...
    <resheader name="resmimetype">text/microsoft-resx</resheader>
    <resheader name="version">2.0</resheader>
    <resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
    <resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
    <data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
    <data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
    <data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
      <value>[base64 mime encoded serialized .NET Framework object]</value>
    </data>
    <data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
      <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
      <comment>This is a comment</comment>

```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

```
mimetype: application/x-microsoft.net.object.binary.base64
value    : The object must be serialized with
          : System.Serialization.Formatters.Binary.BinaryFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.soap.base64
value    : The object must be serialized with
          : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.bytearray.base64
value    : The object must be serialized into a byte array
          : using a System.ComponentModel.TypeConverter
          : and then encoded with base64 encoding.
```

```
-->
```

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="metadata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        <xsd:attribute name="name" type="xsd:string" />
        <xsd:attribute name="type" type="xsd:string" />
        <xsd:attribute name="mimetype" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="assembly">
    <xsd:complexType>
        <xsd:attribute name="alias" type="xsd:string" />
        <xsd:attribute name="name" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="data">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
            <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
        <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
        <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="resheader">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
    <value>text/microsoft-resx</value>
</resheader>
<resheader name="version">
    <value>2.0</value>
</resheader>
<resheader name="reader">

```

```

    <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">
    <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
</root>

```

Settings.Designer.vb (dans *(root directory name)*\VBHelloWorld\My Project):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

```

Option Strict On

Option Explicit On

Namespace My

<Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _

Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings", "11.0.0.0"), _

Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable

_

Partial Friend NotInheritable Class MySettings

Inherits Global.System.Configuration.ApplicationSettingsBase

Private Shared defaultInstance As MySettings =

CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New MySettings), MySettings)

#Region "My.Settings Auto-Save Functionality"

#If _MyType = "WindowsForms" Then

Private Shared addedHandler As Boolean

```

        Private Shared addedHandlerLockObject As New Object

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
        Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
            If My.Application.SaveMySettingsOnExit Then
                My.Settings.Save()
            End If
        End Sub
    #End If
#End Region

Public Shared ReadOnly Property [Default]() As MySettings
    Get

        #If _MyType = "WindowsForms" Then
            If Not addedHandler Then
                SyncLock addedHandlerLockObject
                    If Not addedHandler Then
                        AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                        addedHandler = True
                    End If
                End SyncLock
            End If
        #End If
        Return defaultInstance
    End Get
End Property
End Class
End Namespace

Namespace My

    <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(> _
Friend Module MySettingsProperty

        <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
    Get

```



```
        Return Global.VBHelloWorld.My.MySettings.Default
    End Get
End Property
End Module
End Namespace
```

Settings.settings (dans *(root directory name)\VBHelloWorld\My Project*):

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
  CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

Planifier une intégration AWS CodeBuild

Avant de l'utiliser AWS CodeBuild, vous devez répondre aux questions suivantes :

1. Où est stocké le code source ? CodeBuild prend actuellement en charge la création à partir des fournisseurs de référentiels de code source suivants. Le code source doit contenir un fichier de spécification de génération (buildspec). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Vous pouvez déclarer une spécification de construction dans la définition d'un projet de construction.

Fournisseur du référentiel	Obligatoire	Documentation
CodeCommit	Nom du référentiel. (Facultatif) ID de validation associé au code source.	Consultez ces rubriques dans le Guide de l'utilisateur AWS CodeCommit : Création d'un CodeCommit référentiel Créez un commit dans CodeCommit
Amazon S3	Nom du compartiment d'entrée. Nom d'objet correspondant au fichier ZIP d'entrée de génération qui contient le code source. (Facultatif) ID de version associé au fichier ZIP	Consultez les rubriques suivantes dans le guide de démarrage Amazon S3 : Créez un compartiment Ajout d'un objet dans un compartiment

Fournisseur du référentiel	Obligatoire	Documentation
	d'entrée de génération.	
GitHub	Nom du référentiel. (Facultatif) ID de validation associé au code source.	Consultez cette rubrique sur le site Web GitHub d'aide : Create a repo
Bitbucket	Nom du référentiel. (Facultatif) ID de validation associé au code source.	Consultez cette rubrique sur le site web de la documentation Bitbucket Cloud : "Création d'un référentiel"

2. Quelles commandes de génération devez-vous exécuter et dans quel ordre ? Par défaut, CodeBuild télécharge l'entrée de build depuis le fournisseur que vous spécifiez et télécharge la sortie de build dans le bucket que vous spécifiez. Vous utilisez la spécification de génération pour indiquer comment transformer l'entrée de génération téléchargée en la sortie de génération attendue. Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).
3. Quels sont les environnements d'exécution et les outils dont vous avez besoin pour exécuter la génération ? Par exemple, effectuez-vous une génération pour Java, Ruby, Python ou Node.js ? La génération a-t-elle besoin de Maven ou Ant, ou d'un compilateur pour Java, Ruby ou Python ? La compilation nécessite-t-elle Git AWS CLI, le ou d'autres outils ?

CodeBuild exécute des builds dans des environnements de build qui utilisent des images Docker. Ces images Docker doivent être stockées dans un type de référentiel pris en charge par CodeBuild. Il s'agit notamment du référentiel d'images CodeBuild Docker, de Docker Hub et d'Amazon Elastic Container Registry (Amazon ECR). Pour plus d'informations sur le référentiel d'images CodeBuild Docker, consultez [Images Docker fournies par CodeBuild](#).

4. Avez-vous besoin de AWS ressources qui ne sont pas fournies automatiquement par CodeBuild ? Dans l'affirmative, de quelles politiques de sécurité ces ressources ont-elles besoin ? Par exemple, vous devrez peut-être modifier le rôle CodeBuild de service CodeBuild pour autoriser l'utilisation de ces ressources.
5. Voulez-vous CodeBuild travailler avec votre VPC ? Dans ce cas, vous avez besoin de l'ID VPC, du sous-réseau IDs et du groupe de sécurité pour IDs votre configuration VPC. Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Une fois que vous avez répondu à ces questions, vous devriez disposer des paramètres et des ressources dont vous avez besoin pour exécuter une génération avec succès. Pour exécuter votre génération, vous pouvez :

- Utilisez la AWS CodeBuild console AWS CLI, ou AWS SDKs. Pour de plus amples informations, veuillez consulter [Exécuter les builds manuellement](#).
- Créez ou identifiez un pipeline dans AWS CodePipeline, puis ajoutez une action de génération ou de test demandant de CodeBuild tester automatiquement votre code, d'exécuter votre build, ou les deux. Pour de plus amples informations, veuillez consulter [À utiliser CodeBuild avec CodePipeline](#).

Référence de spécification de construction pour CodeBuild

Cette rubrique fournit des informations de référence importantes sur les fichiers de spécification de génération (buildspec). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Vous pouvez inclure une spécification de construction dans le code source ou vous pouvez définir une spécification de construction lorsque vous créez un projet de construction. Pour plus d'informations sur le fonctionnement des spécifications de génération, consultez [Comment CodeBuild fonctionne](#).

Rubriques

- [Nom de fichier buildspec et emplacement de stockage](#)
- [Syntaxe d'un fichier buildspec](#)
- [Exemple de fichier buildspec](#)
- [Versions de fichier buildspec](#)
- [Référence Batch Build Buildspec](#)

Nom de fichier buildspec et emplacement de stockage

Si vous incluez une spécification de build dans le cadre du code source, par défaut, le fichier de spécification de build doit être nommé `buildspec.yml` et être placé à la racine de votre répertoire source.

Vous pouvez remplacer le nom et l'emplacement par défaut du fichier de spécification de build. Par exemple, vous pouvez effectuer les actions suivantes :

- Utiliser un fichier de spécification de build distinct pour différentes builds dans le même référentiel, comme par exemple `buildspec_debug.yml` et `buildspec_release.yml`.
- Stocker un fichier de spécification de build ailleurs qu'à la racine de votre répertoire source, par exemple `config/buildspec.yml` ou dans un compartiment S3. Le compartiment S3 doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

Vous pouvez spécifier une seule spécification de build pour un projet de build, quel que soit le nom du fichier de spécification de build.

Pour remplacer le nom, l'emplacement, ou les deux, du fichier de spécification de build par défaut, effectuez l'une des actions suivantes :

- Exécutez la `update-project` commande AWS CLI `create-project` or en définissant la `buildspec` valeur sur le chemin du fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR` Vous pouvez également faire l'équivalent avec l'`create-project` opération dans le AWS SDKs. Pour plus d'informations, consultez [Création d'un projet de génération](#) ou [Modifier les paramètres du projet de construction](#).
- Exécutez la AWS CLI `start-build` commande en définissant la `buildspecOverride` valeur sur le chemin du fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR` Vous pouvez également faire l'équivalent avec l'`start-build` opération dans le AWS SDKs. Pour de plus amples informations, veuillez consulter [Exécuter les builds manuellement](#).
- Dans un AWS CloudFormation modèle, définissez la `BuildSpec` propriété de `Source` in a resource of type `AWS::CodeBuild::Project` sur le chemin d'accès au fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR` Pour plus d'informations, consultez la `BuildSpec` propriété dans la [source AWS CodeBuild du projet](#) dans le guide de AWS CloudFormation l'utilisateur.

Syntaxe d'un fichier buildspec

Les fichiers de spécification de build doivent être créés au format [YAML](#).

Si une commande contient un caractère ou une chaîne de caractères qui ne sont pas pris en charge par YAML, vous devez placer la commande entre guillemets (""). La commande suivante est placée entre guillemets car un deux-points (:) suivi d'un espace n'est pas autorisé dans YAML. Dans la commande, le guillemet est échappé (\").

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\",,]//g')"
```

La spécification de build a la syntaxe suivante :

```
version: 0.2
```

```
run-as: Linux-user-name
```

```
env:
```

```
shell: shell-tag
variables:
  key: "value"
  key: "value"
parameter-store:
  key: "value"
  key: "value"
exported-variables:
  - variable
  - variable
secrets-manager:
  key: secret-id:json-key:version-stage:version-id
git-credential-helper: no | yes

proxy:
  upload-artifacts: no | yes
  logs: no | yes

batch:
  fast-fail: false | true
  # build-list:
  # build-matrix:
  # build-graph:
  # build-fanout:

phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
  RETRY-count-regex
  runtime-versions:
    runtime: version
    runtime: version
  commands:
    - command
    - command
  finally:
    - command
    - command

pre_build:
  run-as: Linux-user-name
  on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
  RETRY-count-regex
```

```
commands:
- command
- command
finally:
- command
- command

build:
  run-as: Linux-user-name
  on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
  RETRY-count-regex
  commands:
    - command
    - command
  finally:
    - command
    - command

  post_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
    RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

  reports:
    report-group-name-or-arn:
      files:
        - location
        - location
      base-directory: location
      discard-paths: no | yes
      file-format: report-format

  artifacts:
    files:
      - location
      - location
    name: artifact-name
    discard-paths: no | yes
    base-directory: location
```



```
exclude-paths: excluded paths
enable-symlinks: no | yes
s3-prefix: prefix
secondary-artifacts:
  artifactIdentifier:
    files:
      - location
      - location
    name: secondary-artifact-name
    discard-paths: no | yes
    base-directory: location
  artifactIdentifier:
    files:
      - location
      - location
    discard-paths: no | yes
    base-directory: location
cache:
  key: key
  fallback-keys:
    - fallback-key
    - fallback-key
  action: restore | save
  paths:
    - path
    - path
```

La spécification de build contient les éléments suivants :

version

Mappage obligatoire. Représente la version de la spécification de build. Nous vous recommandons d'utiliser 0.2.

Note

Bien que la version 0.1 soit toujours prise en charge, nous vous recommandons d'utiliser la version 0.2 dans la mesure du possible. Pour de plus amples informations, veuillez consulter [Versions de fichier buildspec](#).

run-as

Séquence facultative. Disponible uniquement pour les utilisateurs Linux. Spécifie un utilisateur Linux qui exécute des commandes dans ce fichier buildspec. `run-as` accorde à l'utilisateur spécifié des autorisations de lecture et d'exécution. Lorsque vous spécifiez `run-as` en haut du fichier buildspec, il s'applique globalement à toutes les commandes. Si vous ne voulez pas spécifier un utilisateur pour toutes les commandes de fichier buildspec, vous pouvez en spécifier un pour les commandes dans une phase à l'aide de `run-as` dans l'un des blocs phases. Si `run-as` n'est pas spécifié, toutes les commandes sont exécutées en tant qu'utilisateur racine.

env

Séquence facultative. Représente les informations pour une ou plusieurs variables d'environnement personnalisées.

Note

Pour protéger les informations sensibles, les informations suivantes sont masquées dans CodeBuild les journaux :

- AWS clé d'accès IDs. Pour plus d'informations, consultez [la section Gestion des clés d'accès pour les utilisateurs IAM](#) dans le guide de l'Gestion des identités et des accès AWS utilisateur.
- Chaînes spécifiées à l'aide du stockage de paramètres. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.
- Chaînes spécifiées à l'aide de AWS Secrets Manager. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

env/ coque

Séquence facultative. Spécifie le shell pris en charge pour les systèmes d'exploitation Linux ou Windows.

Pour les systèmes d'exploitation Linux, les balises shell prises en charge sont les suivantes :

- `bash`
- `/bin/sh`

Pour les systèmes d'exploitation Windows, les balises shell prises en charge sont les suivantes :

- `powershell.exe`
- `cmd.exe`

env/variables

Obligatoire si `env` est spécifié, et que vous voulez définir des variables d'environnement personnalisées en texte brut. Contient un mappage de *key/value* scalaires, où chaque mappage représente une variable d'environnement personnalisée unique en texte brut. *key* est le nom de la variable d'environnement personnalisée et *value* la valeur de cette variable.

Important

Nous déconseillons vivement le stockage de valeurs sensibles dans les variables d'environnement. Les variables d'environnement peuvent être affichées en texte brut à l'aide d'outils tels que la CodeBuild console et le AWS CLI. Pour les valeurs sensibles, nous vous recommandons d'utiliser à la place le mappage `parameter-store` ou `secrets-manager`, comme décrit plus loin dans cette section.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée. Vous pouvez ajouter ou remplacer des variables d'environnement lorsque vous créez une génération. Pour de plus amples informations, veuillez consulter [Exécuter AWS CodeBuild les builds manuellement](#).
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités. Vous pouvez ajouter des variables d'environnement au niveau d'un projet lorsque

vous créez ou modifiez celui-ci. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) et [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).

- La valeur figurant dans la déclaration buildspec a la priorité la plus faible.

env/magasin de paramètres

Obligatoire si cela env est spécifié et si vous souhaitez récupérer des variables d'environnement personnalisées stockées dans Amazon EC2 Systems Manager Parameter Store. Contient un mappage de *key/value* scalars, où chaque mappage représente une variable d'environnement personnalisée unique stockée dans le magasin de paramètres Amazon EC2 Systems Manager. *key* est le nom que vous utiliserez ultérieurement dans vos commandes de compilation pour faire référence à cette variable d'environnement personnalisée, ainsi *value* que le nom de la variable d'environnement personnalisée stockée dans le magasin de paramètres Amazon EC2 Systems Manager. Pour stocker des valeurs sensibles, consultez la section [Stockage des paramètres de Systems Manager](#) et [procédure pas à pas : création et test d'un paramètre de chaîne \(console\)](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Important

CodeBuild Pour permettre de récupérer des variables d'environnement personnalisées stockées dans Amazon EC2 Systems Manager Parameter Store, vous devez ajouter `ssm:GetParameters` l'action à CodeBuild votre rôle de service. Pour de plus amples informations, veuillez consulter [CodeBuild Autoriser l'interaction avec d'autres AWS services](#).

Toutes les variables d'environnement que vous récupérez depuis Amazon EC2 Systems Manager Parameter Store remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec une valeur `my_value`, et que vous récupérez une variable d'environnement nommée `MY_VAR` avec une valeur `other_value`, la valeur `my_value` est alors remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec une valeur `/usr/local/sbin:/usr/local/bin`, et que vous récupérez une variable d'environnement nommée `PATH` avec une valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est alors remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne stockez pas de variables d'environnement avec un nom commençant par CODEBUILD_. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée. Vous pouvez ajouter ou remplacer des variables d'environnement lorsque vous créez une génération. Pour de plus amples informations, veuillez consulter [Exécuter AWS CodeBuild les builds manuellement](#).
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités. Vous pouvez ajouter des variables d'environnement au niveau d'un projet lorsque vous créez ou modifiez celui-ci. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) et [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).
- La valeur figurant dans la déclaration buildspec a la priorité la plus faible.

env/Secrets Manager

Obligatoire si vous souhaitez récupérer des variables d'environnement personnalisées stockées dans AWS Secrets Manager. Spécifiez un Gestionnaire de Secrets `reference-key` en utilisant le modèle suivant :

`<key>: <secret-id>: <json-key>: <version-stage>: <version-id>`

`<key>`

(Obligatoire) Nom de la variable d'environnement locale. Utilisez ce nom pour accéder à la variable lors de la génération.

`<secret-id>`

(Obligatoire) Le nom ou Amazon Resource Name (ARN) qui sert d'identifiant unique pour le secret. Pour accéder à un secret dans votre compte AWS , spécifiez simplement le nom secret. Pour accéder à un secret dans un autre AWS compte, spécifiez l'ARN du secret.

`<json-key>`

(Facultatif) Spécifie le nom de clé de la paire clé-valeur de Secrets Manager dont vous souhaitez récupérer la valeur. Si vous ne spécifiez pas `json-key`, CodeBuild récupère l'intégralité du texte secret.

<version-stage>

(Facultatif) Spécifie la version secrète que vous souhaitez récupérer à l'aide de l'étiquette intermédiaire attachée à la version. Les étiquettes intermédiaires sont utilisées pour assurer le suivi des différentes versions au cours du processus de rotation. Si vous utilisez la `version-stage`, ne spécifiez pas `version-id`. Si vous ne spécifiez pas d'étape de version ni d'ID de version, par défaut, vous devez récupérer la version avec la valeur d'étape de version de `AWSCURRENT`.

<version-id>

(Facultatif) Spécifie l'identifiant unique de la version du secret que vous souhaitez utiliser. Si vous spécifiez `version-id`, ne spécifiez pas `version-stage`. Si vous ne spécifiez pas d'étape de version ni d'ID de version, par défaut, vous devez récupérer la version avec la valeur d'étape de version de `AWSCURRENT`.

Dans l'exemple suivant, `TestSecret` c'est le nom de la paire clé-valeur stockée dans Secrets Manager. La clé pour `TestSecret` nous `MY_SECRET_VAR`. Vous pouvez accéder à la variable pendant la construction en utilisant le `LOCAL_SECRET_VAR` nom.

```
env:
  secrets-manager:
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

env/variables exportées

Mappage facultatif. Permet de répertorier les variables d'environnement que vous souhaitez exporter. Spécifiez le nom de chaque variable que vous souhaitez exporter sur une ligne distincte sous `exported-variables`. La variable que vous souhaitez exporter doit être disponible dans votre conteneur pendant la génération. La variable que vous exportez peut être une variable d'environnement.

Les variables d'environnement exportées sont utilisées conjointement AWS CodePipeline pour exporter les variables d'environnement de la phase de construction actuelle vers les étapes suivantes du pipeline. Pour plus d'informations, consultez la section [Utilisation des variables](#) dans le Guide de AWS CodePipeline l'utilisateur.

Lors d'une génération, la valeur d'une variable est disponible dès la phase `install`. Elle peut être mise à jour entre le début de la phase `install` et la fin de la phase `post_build`. Lorsque la phase `post_build` est terminée, la valeur des variables exportées ne peut pas changer.

 Note

Les éléments suivants ne peuvent pas être exportés :

- Les secrets du magasin de paramètres Amazon EC2 Systems Manager sont spécifiés dans le projet de construction.
- Secrets Manager : secrets spécifiés dans le projet de construction
- Les variables d'environnement qui commencent par `AWS_`.

`env/ git-credential-helper`

Mappage facultatif. Utilisé pour indiquer s'il CodeBuild utilise son assistant d'identification Git pour fournir des informations d'identification Git. `yess` il est utilisé. Sinon, `no` ou non spécifié. Pour plus d'informations, consultez [gitcredentials](#) sur le site web de Git.

 Note

`git-credential-helper` n'est pas pris en charge pour les générations qui sont déclenchées par un webhook pour un référentiel Git public.

`proxy`

Séquence facultative. Utilisé pour représenter les paramètres si vous exécutez votre génération dans un serveur proxy explicite. Pour de plus amples informations, veuillez consulter [Exécuter CodeBuild sur un serveur proxy explicite](#).

`proxy/charger les artefacts`

Mappage facultatif. Définissez ce paramètre sur `yes` si vous souhaitez que votre build dans un serveur proxy explicite charge des artefacts. La valeur par défaut est `no`.

`proxy/journaux`

Mappage facultatif. Définissez ce `yes` paramètre pour intégrer un serveur proxy explicite afin de créer des CloudWatch journaux. La valeur par défaut est `no`.

phases

Séquence obligatoire. Représente les CodeBuild commandes exécutées pendant chaque phase de la génération.

Note

Dans la version 0.1 de buildspec, CodeBuild exécute chaque commande dans une instance distincte du shell par défaut dans l'environnement de construction. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par conséquent, par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes (par exemple, pour le changement de répertoire ou la définition de variables d'environnement). Pour contourner ce problème, nous vous recommandons d'utiliser la version 0.2, qui permet de résoudre ce problème. Si vous devez utiliser la spécification de build de version 0.1, nous vous recommandons les approches figurant dans [Shells et commandes dans les environnements de génération](#).

phases/*/run-as

Séquence facultative. Utilisez-le dans une phase de build pour spécifier un utilisateur Linux qui exécute ses commandes. Si `run-as` est également spécifié globalement pour toutes les commandes en haut du fichier buildspec, alors l'utilisateur au niveau de phase est prioritaire. Par exemple, si `run-as` spécifie globalement l'utilisateur-1, et pour la phase `install` seule une instruction `run-as` spécifie l'utilisateur 2, alors toutes les commandes dans le fichier buildspec sont exécutées en tant qu'utilisateur-1, sauf les commandes dans la phase `install`, qui sont exécutées en tant qu'utilisateur-2.

phases/*/ en cas de défaillance

Séquence facultative. Spécifie l'action à entreprendre en cas de panne pendant la phase. Il peut s'agir de l'une des valeurs suivantes :

- ABORT- Annulation de la construction.
- CONTINUE- Passez à la phase suivante.
- RETRY- Réessayez la compilation jusqu'à 3 fois avec un message d'erreur correspondant à l'expression `. * régulière`.

- `RETRY-count`- Réessayez la compilation un certain nombre de fois, comme indiqué par un message *count* d'erreur correspondant à l'expression `. *` régulière. Notez qu'*count* doit être compris entre 0 et 100. Par exemple, les valeurs valides incluent `RETRY-4` et `RETRY-8`.
- `RETRY-regex`- Réessayez la compilation jusqu'à 3 fois et utilisez-la *regex* pour inclure une expression régulière correspondant à un message d'erreur spécifié. Par exemple, les valeurs valides incluent `Retry-.*Error: Unable to connect to database.*` et `RETRY-invalid+`.
- `RETRY-count-regex`- Réessayez de compiler un certain nombre de fois, comme indiqué par *count*. Notez qu'*count* doit être compris entre 0 et 100. Vous pouvez également l'utiliser *regex* pour inclure une expression régulière correspondant au message d'erreur. Par exemple, les valeurs valides incluent `Retry-3-.*connection timed out.*` et `RETRY-8-invalid+`.

Si cette propriété n'est pas spécifiée, le processus d'échec suit les phases de transition, comme indiqué dans [Transitions des phases de génération](#).

Important

L'on-failureattribut n'est pas pris en charge lors de l'utilisation du calcul Lambda ou de la capacité réservée. Cet attribut ne fonctionne qu'avec les images de calcul EC2 fournies par CodeBuild.

phases/*/ enfin

Bloc facultatif. Les commandes spécifiées dans un `finally` bloc sont exécutées après les commandes du `commands` bloc. Les commandes d'un `finally` bloc sont exécutées même si une commande du `commands` bloc échoue. Par exemple, si le `commands` bloc contient trois commandes et que la première échoue, CodeBuild ignore les deux commandes restantes et exécute toutes les commandes du `finally` bloc. La phase est réussie lorsque toutes les commandes des blocs `commands` et `finally` s'exécutent avec succès. Si une commande d'une phase échoue, la phase échoue.

Les noms de phase de génération autorisés sont :

phases/installer

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées pendant l'installation. Nous vous recommandons d'utiliser la phase `install` uniquement pour l'installation

de packages dans l'environnement de génération. Par exemple, vous pouvez utiliser cette phase pour installer un framework de test de code tel que Mocha ou RSpec.

phases/installer/versions d'exécution

Séquence facultative. Une version d'exécution est prise en charge avec l'image standard Ubuntu 5.0 ou ultérieure et l'image standard Amazon Linux 2 4.0 ou version ultérieure. Si cette valeur est spécifiée, au moins une exécution doit être incluse dans cette section. Spécifiez un environnement d'exécution utilisant une version spécifique, une version majeure suivie de .x pour spécifier que CodeBuild utilise cette version majeure avec sa dernière version mineure, ou latest pour utiliser les versions majeure et mineure les plus récentes (par exemple `ruby: 3.2`, `nodejs: 18.x`, ou `java: latest`). Vous pouvez spécifier l'exécution à l'aide d'un nombre ou d'une variable d'environnement. Par exemple, si vous utilisez l'image standard 4.0 d'Amazon Linux 2, ce qui suit indique que la version 17 de Java, la dernière version mineure de python version 3 et une version contenue dans une variable d'environnement de Ruby sont installées. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

Vous pouvez spécifier un ou plusieurs runtimes dans la section `runtime-versions` de votre fichier `buildspec`. Si votre runtime dépend d'un autre runtime, vous pouvez également spécifier son runtime dépendant dans le fichier `buildspec`. Si vous ne spécifiez aucun environnement d'exécution dans le fichier `buildspec`, CodeBuild choisit les environnements d'exécution par défaut disponibles dans l'image que vous utilisez. Si vous spécifiez un ou plusieurs environnements d'exécution, utilisez uniquement CodeBuild ces environnements d'exécution. Si aucun environnement d'exécution dépendant n'est spécifié, CodeBuild tente de le choisir pour vous.

Si deux exécutions spécifiées entrent en conflit, la génération échoue. Par exemple, `android: 29` et `java: openjdk11` sont en conflit, par conséquent si les deux sont spécifiés, la génération échoue.

Pour plus d'informations sur les environnements d'exécution disponibles, consultez [Runtimes disponibles](#).

 Note

Si vous spécifiez une `runtime-versions` section et utilisez une image autre qu'Ubuntu Standard Image 2.0 ou version ultérieure, ou l'image standard Amazon Linux 2 (AL2) 1.0 ou version ultérieure, le build émet l'avertissement « `»Skipping install of runtimes. Runtime version selection is not supported by this build image.` »

phases/installer/commandes

Séquence facultative. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée pendant l'installation. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/prégénération

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées avant la génération. Par exemple, vous pouvez utiliser cette phase pour vous connecter à Amazon ECR, ou vous pouvez installer des dépendances npm.

phases/prégénération/commandes

Séquence obligatoire si `pre_build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée avant la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/génération

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées pendant la génération. Par exemple, vous pouvez utiliser cette phase pour exécuter Mocha ou sbt. RSpec

phases/génération/commandes

Obligatoire si `build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée pendant la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/post-génération

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées après la génération. Par exemple, vous pouvez utiliser Maven pour empaqueter les artefacts de

construction dans un fichier JAR ou WAR, ou vous pouvez transférer une image Docker dans Amazon ECR. Vous pouvez ensuite envoyer une notification de build via Amazon SNS.

phases/post-génération/commandes

Obligatoire si cela `post_build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée après la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

rapports

report-group-name-or-fil

Séquence facultative. Spécifie le groupe de rapports auquel les rapports sont envoyés. Un projet peut comporter un maximum de cinq groupes de rapports. Spécifiez l'ARN d'un groupe de rapports existant ou le nom d'un nouveau groupe de rapports. Si vous spécifiez un nom, CodeBuild crée un groupe de rapports en utilisant le nom de votre projet et le nom que vous spécifiez dans le format `<project-name>-<report-group-name>`. Le nom du groupe de rapports peut également être défini à l'aide d'une variable d'environnement dans la spécification de construction, telle que `$REPORT_GROUP_NAME`. Pour de plus amples informations, veuillez consulter [Attribution des noms des groupes de rapports](#).

rapports/<groupe de rapports>/fichiers

Séquence obligatoire. Représente les emplacements qui contiennent les données brutes des résultats de test générés par le rapport. Contient une séquence de scalaires, chaque scalaire représentant un emplacement distinct où se CodeBuild trouvent les fichiers de test, par rapport à l'emplacement de construction d'origine ou, s'il est défini, au `base-directory`. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-test-report-file.json`).
- Un fichier unique dans un sous-répertoire (par exemple, `my-subdirectory/my-test-report-file.json` ou `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/*'` représente tous les fichiers de manière récursive.
- `my-subdirectory/*` représente tous les fichiers d'un sous-répertoire nommé `my-subdirectory`.

- *my-subdirectory*/**/*représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé. *my-subdirectory*

rapports/<groupe de rapports>/format de fichier

Mappage facultatif. Représente le format du fichier de rapport. Si non spécifié, JUNITXML est utilisé. Cette valeur ne distingue pas les majuscules et minuscules. Les valeurs possibles sont :

Rapports d'essais

CUCUMBERJSON

Cucumber JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

NUnit 3 XML

TESTNGXML

TestNG XML

VISUALSTUDIOTRX

Visual Studio TRX

Rapports de couverture du code

CLOVERXML

Clover XML

COBERTURAXML

Cobertura XML

JACOCOXML

JaCoCo XML

SIMPLECOV

SimpleCov JSON

Note

CodeBuild [accepte les rapports de couverture du code JSON générés par simplecov, et non par simplecov-json.](#)

rapports/<groupe de rapports>/répertoire de base

Mappage facultatif. Représente un ou plusieurs répertoires de niveau supérieur, relatifs à l'emplacement de construction d'origine, qui CodeBuild permettent de déterminer où trouver les fichiers de test bruts.

rapports/<groupe de rapports>/annuler les chemins

Facultatif. Spécifie si les répertoires de fichiers de rapport sont aplatis dans la sortie. Si cela n'est pas spécifié ou si no est défini, les fichiers de rapport sont générés avec leur structure de répertoire intacte. Si yes est défini, tous les fichiers de test sont placés dans le même répertoire de sortie. Par exemple, si un chemin d'accès à un résultat de test est com/myapp/mytests/TestResult.xml, la définition de yes place ce fichier dans /TestResult.xml.

artefacts

Séquence facultative. Représente des informations indiquant où se CodeBuild trouve la sortie de génération et comment CodeBuild la préparer pour le téléchargement vers le compartiment de sortie S3. Cette séquence n'est pas obligatoire si, par exemple, vous créez et transférez une image Docker vers Amazon ECR, ou si vous exécutez des tests unitaires sur votre code source, mais que vous ne le créez pas.

Note

Les métadonnées Amazon S3 ont un CodeBuild en-tête nommé x-amz-meta-codebuild-buildarn qui contient le nom buildArn de la CodeBuild version qui publie les artefacts sur Amazon S3. Le buildArn est ajouté pour permettre le suivi de la source des notifications et pour indiquer la version à partir de laquelle l'artefact est généré.

artefacts/fichiers

Séquence obligatoire. Représente les emplacements qui contiennent les artefacts de sortie de génération dans l'environnement de génération. Contient une séquence de scalaires, où chaque scalaire représente un emplacement distinct où CodeBuild peut trouver des artefacts de sortie de génération, par rapport à l'emplacement de génération d'origine ou, si défini, au répertoire de base. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-file.jar`).
- Un fichier unique dans un sous-répertoire (par exemple, `my-subdirectory/my-file.jar` ou `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` représente tous les fichiers de manière récursive.
- `my-subdirectory/*` représente tous les fichiers d'un sous-répertoire nommé `my-subdirectory`.
- `my-subdirectory/**/*` représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé `my-subdirectory`.

Lorsque vous spécifiez les emplacements des artefacts de sortie de construction, vous CodeBuild pouvez localiser l'emplacement de construction d'origine dans l'environnement de construction. Vous n'avez pas besoin de préfixer les emplacements d'artefact de sortie de génération avec le chemin de l'emplacement de génération d'origine, ni de spécifier `./` ou un élément similaire. Si vous souhaitez connaître le chemin d'accès à cet emplacement, vous pouvez exécuter une commande comme `echo $CODEBUILD_SRC_DIR` pendant une génération. L'emplacement de chaque environnement de génération peut être légèrement différent.

artefacts/nom

Nom facultatif. Spécifie un nom pour votre artefact de génération. Ce nom est utilisé lorsque l'une des conditions suivantes est vraie.

- Vous utilisez l' API CodeBuild pour créer vos versions et l'`overrideArtifactName`indicateur est défini sur l'`ProjectArtifact`objet lorsqu'un projet est mis à jour, qu'un projet est créé ou qu'un build est lancé.
- Vous utilisez la CodeBuild console pour créer vos versions, un nom est spécifié dans le fichier `buildspec` et vous sélectionnez Activer le versionnement sémantique lorsque vous créez ou mettez à jour un projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).

Vous pouvez spécifier un nom dans le fichier de spécification de build qui est calculé au moment de la génération. Le nom spécifié dans un fichier de spécification de build utilise le langage de

commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour de plus amples informations, veuillez consulter [Langage de commandes Shell](#).

- Ceci est un exemple de nom d'artefact suivi de la date à laquelle l'artefact a été créé.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

- Il s'agit d'un exemple de nom d'artefact qui utilise une variable d' CodeBuild environnement. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

- Il s'agit d'un exemple de nom d'artefact qui utilise une variable d' CodeBuild environnement à laquelle est ajoutée la date de création de l'artefact.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: $AWS_REGION-$(date +%Y-%m-%d)
```


Vous pouvez ajouter des informations de chemin au nom afin que les artefacts nommés soient placés dans des répertoires en fonction du chemin indiqué dans le nom. Dans cet exemple, les artefacts de construction sont placés dans la sortie sous `builds/<build number>/my-artifacts`.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: builds/${CODEBUILD_BUILD_NUMBER}/my-artifacts
```

artefacts/ignorer les chemins

Facultatif. Spécifie si les répertoires d'artefacts de génération sont aplatis dans la sortie. Si cela n'est pas spécifié, si `no` est défini, les artefacts de génération sont générés avec leur structure de répertoire intacte. Si `yes` est défini, tous les artefacts de génération sont placés dans le même répertoire de sortie. Par exemple, si un chemin d'accès à un fichier dans l'artefact de sortie de génération est `com/mycompany/app/HelloWorld.java`, la définition de `yes` permet de placer ce fichier dans `/HelloWorld.java`.

artefacts/répertoire de base

Mappage facultatif. Représente un ou plusieurs répertoires de niveau supérieur, relatifs à l'emplacement de génération d'origine, qui permettent CodeBuild de déterminer les fichiers et sous-répertoires à inclure dans l'artefact de sortie de génération. Les valeurs valides sont les suivantes :

- Un répertoire unique de niveau supérieur (par exemple, `my-directory`).
- `'my-directory*'` représente tous les répertoires de niveau supérieur avec les noms commençant par `my-directory`.

Les répertoires de niveau supérieur correspondants ne sont pas inclus dans l'artefact de sortie de génération, seulement leurs fichiers et sous-répertoires.

Vous pouvez utiliser `files` et `discard-paths` pour limiter encore plus les fichiers et sous-répertoires inclus. Par exemple, pour la structure de répertoire suivante :

```
.  
### my-build-1  
#   ### my-file-1.txt  
### my-build-2  
    ### my-file-2.txt  
    ### my-subdirectory  
        ### my-file-3.txt
```

Et pour la séquence artifacts suivante :

```
artifacts:  
  files:  
    - '*/my-file-3.txt'  
  base-directory: my-build-2
```

Le sous-répertoire et le fichier suivants seront inclus dans l'artefact de sortie de génération :

```
.  
### my-subdirectory  
    ### my-file-3.txt
```

Alors que pour la séquence artifacts suivante :

```
artifacts:  
  files:  
    - '**/*'  
  base-directory: 'my-build*'  
  discard-paths: yes
```

Les fichiers suivants seront inclus dans l'artefact de sortie de génération :

```
.  
### my-file-1.txt  
### my-file-2.txt  
### my-file-3.txt
```

artefacts/ chemins d'exclusion

Mappage facultatif. Représente un ou plusieurs chemins, relatifs à `base-directory`, qui CodeBuild seront exclus de la construction des artefacts. L'astérisque (*) correspond à zéro ou

plusieurs caractères d'un composant de nom sans dépasser les limites d'un dossier. Un double astérisque (******) correspond à zéro ou plusieurs caractères d'un composant de nom dans tous les répertoires.

Voici des exemples de chemins d'exclusion :

- Pour exclure un fichier de tous les répertoires : `"/file-name/"`
- Pour exclure tous les dossiers à points : `"/.*/"`
- Pour exclure tous les fichiers à points : `"/.*"`

artefacts/ activer les liens symboliques

Facultatif. Si le type de sortie est ZIP, indique si les liens symboliques internes sont conservés dans le fichier ZIP. Si tel est le cas, tous les liens symboliques internes de la source seront conservés dans le fichier ZIP des artefacts.

artefacts/ préfixe s3

Facultatif. Spécifie un préfixe utilisé lorsque les artefacts sont envoyés vers un compartiment Amazon S3 et que le type d'espace de noms est. BUILD_ID Lorsque'il est utilisé, le chemin de sortie dans le compartiment est `<s3-prefix>/<build-id>/<name>.zip`.

artefacts/artefacts secondaires

Séquence facultative. Représente une ou plusieurs définitions d'artefacts comme correspondance entre l'identificateur d'un artefact et la définition d'un artefact. Chaque identifiant d'artefact de ce bloc doit correspondre à un artefact défini dans l'attribut `secondaryArtifacts` de votre projet. Chaque définition distincte a la même syntaxe que le bloc `artifacts` ci-dessus.

 Note

La [artifacts/files](#) séquence est toujours requise, même lorsque seuls des artefacts secondaires sont définis.

Par exemple, si la structure de votre projet est la suivante :

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
```

```
    "type": "S3",
    "location": "<output-bucket1>",
    "artifactIdentifier": "artifact1",
    "name": "secondary-artifact-name-1"
  },
  {
    "type": "S3",
    "location": "<output-bucket2>",
    "artifactIdentifier": "artifact2",
    "name": "secondary-artifact-name-2"
  }
]
```

Votre fichier buildspec ressemble alors à ceci :

```
version: 0.2

phases:
build:
  commands:
    - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      files:
        - directory/file1
      name: secondary-artifact-name-1
    artifact2:
      files:
        - directory/file2
      name: secondary-artifact-name-2
```

cache

Séquence facultative. Représente des informations sur l'endroit où CodeBuild préparer les fichiers pour le téléchargement du cache dans un compartiment de cache S3. Cette séquence n'est pas requise si le type de cache du projet est No Cache.

cache/clé

Séquence facultative. Représente la clé primaire utilisée lors de la recherche ou de la restauration d'un cache. CodeBuild correspond exactement à la clé primaire.

Voici un exemple de clé :

```
key: npm-key-${codebuild-hash-files package-lock.json} }
```

clés de cache ou de secours

Séquence facultative. Représente une liste de clés de secours utilisées de manière séquentielle lorsqu'un cache est introuvable à l'aide de la clé primaire. Jusqu'à cinq clés de secours sont prises en charge, et chacune est mise en correspondance à l'aide d'une recherche par préfixe. Cette séquence sera ignorée si la clé n'est pas fournie.

Voici un exemple pour les clés de secours :

```
fallback-keys:
  - npm-key-${codebuild-hash-files package-lock.json} }
  - npm-key-
  - npm-
```

cache/action

Séquence facultative. Spécifie l'action à effectuer sur le cache. Les valeurs valides sont les suivantes :

- `restore` qui restaure uniquement le cache sans enregistrer les mises à jour.
- `save` qui enregistre uniquement le cache sans restaurer une version précédente.

Si aucune valeur n'est fournie, la restauration et la sauvegarde sont effectuées CodeBuild par défaut.

cache/chemins

Séquence obligatoire. Représente les emplacements du cache. Contient une séquence de scalaires, chaque scalaire représentant un emplacement distinct où CodeBuild peuvent trouver les artefacts de sortie de construction, par rapport à l'emplacement de construction d'origine ou, s'il est défini, au répertoire de base. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-file.jar`).

- Un fichier unique dans un sous-répertoire (par exemple, *my-subdirectory*/my-file.jar ou *my-parent-subdirectory*/my-subdirectory/my-file.jar).
- `'**/*'` représente tous les fichiers de manière récursive.
- *my-subdirectory*/* représente tous les fichiers d'un sous-répertoire nommé *my-subdirectory*.
- *my-subdirectory*/**/* représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé *my-subdirectory*.

Important

Comme une déclaration de spécification de build doit être dans un format YAML valide, les espaces sont importants dans une déclaration de spécification de build. Si le nombre d'espaces dans votre déclaration de spécification de build n'est pas valide, les builds peuvent échouer immédiatement. Vous pouvez utiliser un validateur YAML pour tester si vos déclarations de spécification de build sont dans un format YAML valide.

Si vous utilisez le AWS CLI, ou le AWS SDKs pour déclarer une spécification de construction lorsque vous créez ou mettez à jour un projet de construction, la spécification de construction doit être une chaîne unique exprimée au format YAML, avec les espaces blancs et les caractères d'échappement de nouvelle ligne requis. Vous trouverez un exemple dans la section suivante.

Si vous utilisez les AWS CodePipeline consoles CodeBuild ou au lieu d'un fichier buildspec.yml, vous pouvez insérer des commandes pour la phase uniquement. build Au lieu d'utiliser la syntaxe précédente, vous répertoriez, sur une seule ligne, toutes les commandes que vous souhaitez exécuter lors de la phase de génération (build). Pour plusieurs commandes, séparez celles-ci avec && (par exemple, mvn test && mvn package).

Vous pouvez utiliser les CodePipeline consoles CodeBuild ou au lieu d'un fichier buildspec.yml pour spécifier les emplacements des artefacts de sortie de génération dans l'environnement de génération. Au lieu d'utiliser la syntaxe précédente, vous répertoriez, sur une seule ligne, tous les emplacements. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, buildspec.yml, target/my-app.jar).

Exemple de fichier buildspec

Voici un exemple de fichier buildspec.yml.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`

reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
    files:
      - "**/*"
    base-directory: 'target/tests/reports'
    discard-paths: no
  reportGroupCucumberJson:
    files:
      - 'cucumber/target/cucumber-tests.xml'
```

```

    discard-paths: yes
    file-format: CUCUMBERJSON # default is JUNITXML
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
  secondary-artifacts:
    artifact1:
      files:
        - target/artifact-1.0.jar
      discard-paths: yes
    artifact2:
      files:
        - target/artifact-2.0.jar
      discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'

```

Voici un exemple de la spécification de construction précédente, exprimée sous forme de chaîne unique, à utiliser avec le AWS CLI, ou le AWS SDKs

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-
amd64\n\nparameter-store:\n  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n
phases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n
- apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This
always runs even if the update or install command fails\n  pre_build:\n    commands:
\n      - echo Entered the pre_build phase...\n      - docker login -u User -p
$LOGIN_PASSWORD\n    finally:\n      - echo This always runs even if the login command
fails\n  build:\n    commands:\n      - echo Entered the build phase...\n      - echo
Build started on `date`\n      - mvn install\n    finally:\n      - echo This always
runs even if the install command fails\n  post_build:\n    commands:\n      - echo
Entered the post_build phase...\n      - echo Build completed on `date`\n\nreports:
\n  reportGroupJUnitXml:\n    files:\n      - \"**/*\"\n    base-directory: 'target/
tests/reports'\n    discard-paths: false\n  reportGroupCucumberJson:\n    files:\n
- 'cucumber/target/cucumber-tests.xml'\n    file-format: CUCUMBERJSON\n\nartifacts:\n
  files:\n    - target/messageUtil-1.0.jar\n    discard-paths: yes\n  secondary-artifacts:
\n    artifact1:\n      files:\n        - target/messageUtil-1.0.jar\n      discard-
paths: yes\n    artifact2:\n      files:\n        - target/messageUtil-1.0.jar\n
      discard-paths: yes\n  cache:\n    paths:\n      - '/root/.m2/**/*'

```

Voici un exemple des commandes de la build phase, à utiliser avec les CodePipeline consoles CodeBuild or.


```
echo Build started on `date` && mvn install
```

Dans les exemples suivants :

- Une variable d'environnement personnalisée, en texte brut, avec la clé `JAVA_HOME` et la valeur `/usr/lib/jvm/java-8-openjdk-amd64` est définie.
- Une variable d'environnement personnalisée nommée `dockerLoginPassword` stockée dans Amazon EC2 Systems Manager Parameter Store est référencée ultérieurement dans les commandes de construction à l'aide de la `LOGIN_PASSWORD` clé.
- Vous ne pouvez pas modifier ces noms de phase de génération. Les commandes exécutées dans cet exemple sont `apt-get update -y` et `apt-get install -y maven` (pour installer Apache Maven), `mvn install` (pour compiler, tester et emballer le code source dans un artefact de sortie de build et pour installer l'artefact de sortie de build dans son référentiel interne), `docker login` (pour se connecter à Docker avec le mot de passe correspondant à la valeur de la variable d'environnement personnalisée que `dockerLoginPassword` vous avez définie dans Amazon EC2 Systems Manager Parameter Store) et plusieurs commandes. `echo` Les commandes sont incluses ici pour montrer comment les CodeBuild commandes sont exécutées et l'ordre dans lequel elles sont exécutées.
- `files` représente les fichiers à charger dans l'emplacement de sortie de génération. Dans cet exemple, CodeBuild télécharge le fichier `messageUtil-1.0.jar` unique. Le fichier `messageUtil-1.0.jar` peut être trouvé dans le répertoire relatif nommé `target` dans l'environnement de génération. Comme `discard-paths: yes` est spécifié, `messageUtil-1.0.jar` est chargé directement (et non dans un répertoire `target` intermédiaire). Le nom de fichier `messageUtil-1.0.jar` et le nom de répertoire relatif de `target` sont basés sur la façon dont Apache crée et stocke les artefacts de sortie de génération pour cet exemple uniquement. Dans vos propres scénarios, ces noms de fichier et ces répertoires seront différents.
- `reports` représente deux groupes de rapports qui génèrent des rapports pendant la génération :
 - `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` spécifie l'ARN d'un groupe de rapports. Les résultats des tests générés par le framework de test se trouvent dans le répertoire `target/tests/reports`. Le format de fichier est `JUnitXml` et le chemin d'accès n'est pas supprimé des fichiers contenant les résultats du test.
 - `reportGroupCucumberJson` spécifie un nouveau groupe de rapports. Si le nom du projet est `my-project`, un groupe de rapports portant le nom `my-project-`

`reportGroupCucumberJson` est créé lors de l'exécution d'une génération. Les résultats des tests générés par le framework de test sont dans `cucumber/target/cucumber-tests.xml`. Le format de fichier test est `CucumberJson` et le chemin d'accès est supprimé des fichiers contenant les résultats du test.

Versions de fichier buildspec

Le tableau suivant répertorie les versions de spécification de build et les modifications entre les versions.

Version	Modifications
0.2	• <code>environment_variables</code> a été renommée en <code>env</code>. • <code>plaintext</code> a été renommée en <code>variables</code>. • La propriété <code>type</code> pour <code>artifacts</code> est obsolète. • Dans la version 0.1, AWS CodeBuild exécute chaque commande de construction dans une instance distincte du shell par défaut dans l'environnement de construction. Dans la version 0.2, CodeBuild exécute toutes les commandes de construction dans la même instance du shell par défaut dans l'environnement de construction.
0.1	Il s'agit de la définition initiale du format de spécification de génération.

Référence Batch Build Buildspec

Cette rubrique contient la référence buildspec pour les propriétés de construction par lots.

lot

Mappage facultatif. Les paramètres de génération par lots pour le projet.

batch ou échec rapide

Facultatif. Spécifie le comportement de la génération par lots lorsqu'une ou plusieurs tâches de génération échouent.

false

La valeur par défaut. Toutes les versions en cours seront terminées.

true

Toutes les versions en cours seront arrêtées en cas d'échec de l'une des tâches de génération.

Par défaut, toutes les tâches de génération par lots sont exécutées avec les paramètres de génération tels que env et phases, spécifiés dans le fichier buildspec. Vous pouvez remplacer les paramètres de construction par défaut en spécifiant des env valeurs différentes ou un fichier de spécification de construction différent dans le paramètre. batch/<batch-type>/buildspec

Le contenu de la batch propriété varie en fonction du type de construction par lots spécifié. Les types de construction par lots possibles sont les suivants :

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)
- [batch/build-fanout](#)

batch/build-graph

Définit un graphe de construction. Un graphe de génération définit un ensemble de tâches qui dépendent d'autres tâches du lot. Pour de plus amples informations, veuillez consulter [Construire un graphe](#).

Cet élément contient un ensemble de tâches de construction. Chaque tâche de génération contient les propriétés suivantes.

identifiant

Obligatoire. Identifiant de la tâche.

build spec

Facultatif. Le chemin et le nom du fichier buildspec à utiliser pour cette tâche. Si ce paramètre n'est pas spécifié, le fichier buildspec actuel est utilisé.

session de débogage

Facultatif. Valeur booléenne qui indique si le débogage de session est activé pour cette génération par lots. Pour plus d'informations sur le débogage de session, consultez [Déboguer les builds avec le gestionnaire de session](#).

false

Le débogage de session est désactivé.

true

Le débogage de session est activé.

dépendre

Facultatif. Tableau d'identificateurs de tâches dont dépend cette tâche. Cette tâche ne sera pas exécutée tant que ces tâches ne seront pas terminées.

env

Facultatif. L'environnement de génération remplace la tâche. Cela peut contenir les propriétés suivantes :

type de calcul

Identifiant du type de calcul à utiliser pour la tâche. Voir ComputeType [the section called “Modes et types de calcul de l'environnement de création”](#) in pour les valeurs possibles.

flotte

Identifiant de la flotte à utiliser pour la tâche. Pour plus d'informations, consultez [the section called “Exécutez des builds sur des flottes à capacité réservée”](#).

image

Identifiant de l'image à utiliser pour la tâche. Voir Identifiant de l'image dans [the section called “Images Docker fournies par CodeBuild”](#) pour les valeurs possibles.

mode privilégié

Valeur booléenne qui indique s'il faut exécuter le démon Docker dans un conteneur Docker. Défini sur `true` uniquement si le projet de génération est utilisé pour créer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est `false`.

type

Identifiant du type d'environnement à utiliser pour la tâche. Voir Type d'environnement [the section called “Modes et types de calcul de l'environnement de création”](#) pour les valeurs possibles.

variables

Les variables d'environnement qui seront présentes dans l'environnement de construction. Pour plus d'informations, consultez [env/variables](#).

Note

Notez que le type de calcul et le parc ne peuvent pas être fournis dans le même identifiant pour une seule version.

ignorer l'échec

Facultatif. Une valeur booléenne qui indique si l'échec de cette tâche de génération peut être ignoré.

false

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

true

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

Voici un exemple d'entrée buildspec du graphe de construction :

```
batch:
  fast-fail: false
```

```
build-graph:
  - identifier: build1
    env:
      variables:
        BUILD_ID: build1
    ignore-failure: false
  - identifier: build2
    buildspec: build2.yml
    env:
      variables:
        BUILD_ID: build2
    depend-on:
      - build1
  - identifier: build3
    env:
      variables:
        BUILD_ID: build3
    depend-on:
      - build2
  - identifier: build4
    env:
      compute-type: ARM_LAMBDA_1GB
  - identifier: build5
    env:
      fleet: fleet_name
```

batch/build-list

Définit une liste de build. Une liste de construction est utilisée pour définir un certain nombre de tâches exécutées en parallèle. Pour de plus amples informations, veuillez consulter [Construire une liste](#).

Cet élément contient un ensemble de tâches de construction. Chaque tâche de génération contient les propriétés suivantes.

identifiant

Obligatoire. Identifiant de la tâche.

build spec

Facultatif. Le chemin et le nom du fichier buildspec à utiliser pour cette tâche. Si ce paramètre n'est pas spécifié, le fichier buildspec actuel est utilisé.

session de débogage

Facultatif. Valeur booléenne qui indique si le débogage de session est activé pour cette génération par lots. Pour plus d'informations sur le débogage de session, consultez [Déboguer les builds avec le gestionnaire de session](#).

`false`

Le débogage de session est désactivé.

`true`

Le débogage de session est activé.

env

Facultatif. L'environnement de génération remplace la tâche. Cela peut contenir les propriétés suivantes :

type de calcul

Identifiant du type de calcul à utiliser pour la tâche. Voir ComputeType [the section called “Modes et types de calcul de l'environnement de création”](#) in pour les valeurs possibles.

flotte

Identifiant de la flotte à utiliser pour la tâche. Pour plus d'informations, consultez [the section called “Exécutez des builds sur des flottes à capacité réservée”](#).

image

Identifiant de l'image à utiliser pour la tâche. Voir Identifiant de l'image dans [the section called “Images Docker fournies par CodeBuild”](#) pour les valeurs possibles.

mode privilégié

Valeur booléenne qui indique s'il faut exécuter le démon Docker dans un conteneur Docker. Défini sur `true` uniquement si le projet de génération est utilisé pour créer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est `false`.

type

Identifiant du type d'environnement à utiliser pour la tâche. Voir Type d'environnement [the section called “Modes et types de calcul de l'environnement de création”](#) pour les valeurs possibles.

variables

Les variables d'environnement qui seront présentes dans l'environnement de construction. Pour plus d'informations, consultez [env/variables](#).

Note

Notez que le type de calcul et le parc ne peuvent pas être fournis dans le même identifiant pour une seule version.

ignorer l'échec

Facultatif. Une valeur booléenne qui indique si l'échec de cette tâche de génération peut être ignoré.

false

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

true

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

Voici un exemple d'entrée buildspec de liste de construction :

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
```



```
compute-type: ARM_LAMBDA_1GB
- identifier: build4
  env:
    fleet: fleet_name
- identifier: build5
  env:
    compute-type: GENERAL_LINUX_XLAGRE
```

batch/build-matrix

Définit une matrice de construction. Une matrice de génération définit les tâches avec différentes configurations qui s'exécutent en parallèle. CodeBuild crée une version distincte pour chaque combinaison de configuration possible. Pour de plus amples informations, veuillez consulter [Construire une matrice](#).

statique

Les propriétés statiques s'appliquent à toutes les tâches de génération.

ignorer l'échec

Facultatif. Une valeur booléenne qui indique si l'échec de cette tâche de génération peut être ignoré.

false

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

true

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

env

Facultatif. L'environnement de construction prévaut pour toutes les tâches.

mode privilégié

Valeur booléenne qui indique s'il faut exécuter le démon Docker dans un conteneur Docker. Défini sur true uniquement si le projet de génération est utilisé pour créer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est false.

type

Identifiant du type d'environnement à utiliser pour la tâche. Voir Type d'environnement [the section called “Modes et types de calcul de l'environnement de création”](#) pour les valeurs possibles.

dynamique

Les propriétés dynamiques définissent la matrice de construction.

build spec

Facultatif. Tableau contenant le chemin et les noms des fichiers buildspec à utiliser pour ces tâches. Si ce paramètre n'est pas spécifié, le fichier buildspec actuel est utilisé.

env

Facultatif. L'environnement de génération prévaut pour ces tâches.

type de calcul

Tableau contenant les identificateurs des types de calcul à utiliser pour ces tâches. Voir ComputeType [the section called “Modes et types de calcul de l'environnement de création”](#) in pour les valeurs possibles.

image

Tableau contenant les identifiants des images à utiliser pour ces tâches. Voir Identifiant de l'image dans [the section called “Images Docker fournies par CodeBuild”](#) pour les valeurs possibles.

variables

Un tableau qui contient les variables d'environnement qui seront présentes dans les environnements de génération pour ces tâches. Pour plus d'informations, consultez [env/variables](#).

Voici un exemple d'entrée buildspec de matrice de construction :

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
```

```
buildspec:
  - matrix1.yml
  - matrix2.yml
env:
  variables:
    MY_VAR:
      - VALUE1
      - VALUE2
      - VALUE3
```

Pour de plus amples informations, veuillez consulter [Construire une matrice](#).

batch/build-fanout

Définit un ventilateur de construction. Un fanout de construction est utilisé pour définir une tâche divisée en plusieurs versions exécutées en parallèle. Pour de plus amples informations, veuillez consulter [Exécuter des tests parallèles dans des builds par lots](#).

Cet élément contient une tâche de génération qui peut être divisée en plusieurs versions. La build-fanout section contient les propriétés suivantes.

parallélisme

Obligatoire. Le nombre de versions qui exécuteront des tests en parallèle.

ignorer l'échec

Facultatif. Une valeur booléenne qui indique si l'échec de l'une des tâches de génération du fanout peut être ignoré. Cette valeur d'ignore-failure sera appliquée à toutes les versions de fanout.

false

La valeur par défaut. Si une tâche de génération de fanout échoue, la génération par lots échouera.

true

Si une tâche de génération de fanout échoue, la génération par lots peut toujours réussir.

Voici un exemple d'entrée buildspec de build fanout :

```
version: 0.2
```

```
batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'equal-distribution'
```

Pour plus d'informations, consultez [Construire un ventilateur](#) et [Utilisez la codebuild-tests-run commande CLI](#).

Créez une référence d'environnement pour AWS CodeBuild

Lorsque vous appelez AWS CodeBuild pour exécuter un build, vous devez fournir des informations sur l'environnement de build. Un environnement de génération représente une combinaison de système d'exploitation, d'environnement d'exécution d'un langage de programmation et d'outils CodeBuild utilisés pour exécuter une compilation. Pour plus d'informations sur le fonctionnement d'un environnement de génération, consultez [Comment CodeBuild fonctionne](#).

Un environnement de génération contient une image Docker. Pour plus d'informations, consultez [le glossaire Docker](#) sur le site web Docker Docs.

Lorsque vous fournissez des informations CodeBuild sur l'environnement de génération, vous spécifiez l'identifiant d'une image Docker dans un type de référentiel pris en charge. Il s'agit notamment du référentiel d'images CodeBuild Docker, des images accessibles au public dans Docker Hub et des référentiels Amazon Elastic Container Registry (Amazon ECR) auxquels votre compte est AWS autorisé à accéder.

- Nous vous recommandons d'utiliser les images Docker stockées dans le référentiel d'images CodeBuild Docker, car elles sont optimisées pour être utilisées avec le service. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).
- Pour obtenir l'identificateur d'une image Docker disponible publiquement dans Docker Hub, veuillez consulter [Searching for Repositories](#) sur le site web Docker Docs.
- Pour savoir comment utiliser les images Docker stockées dans les référentiels Amazon ECR de votre AWS compte, consultez [Exemple Amazon ECR](#).

En plus d'un identificateur d'image Docker, vous spécifiez également un ensemble de ressources informatiques qui sont utilisées par l'environnement de génération. Pour de plus amples informations, veuillez consulter [Modes et types de calcul de l'environnement de création](#).

Rubriques

- [Images Docker fournies par CodeBuild](#)
- [Modes et types de calcul de l'environnement de création](#)
- [Shells et commandes dans les environnements de génération](#)
- [Variables d'environnement dans les environnements de génération](#)
- [Tâches en arrière-plan dans les environnements de génération](#)

Images Docker fournies par CodeBuild

Une image prise en charge est la dernière version majeure d'une image disponible dans CodeBuild et est mise à jour avec des mises à jour mineures et des correctifs. CodeBuild optimise la durée de provisionnement des versions avec des images prises en charge en les mettant en cache dans l'[Amazon Machine Images \(AMI\) de la machine](#). Si vous souhaitez bénéficier de la mise en cache et minimiser la durée de mise en service de votre build, sélectionnez Toujours utiliser la dernière image pour cette version d'exécution dans la section Version image de la CodeBuild console au lieu d'une version plus détaillée, telle que `aws/codebuild/amazonlinux-x86_64-standard:4.0-1.0.0`

Rubriques

- [Obtenir la liste des images Docker actuelles](#)
- [EC2 calculer des images](#)
- [Images de calcul Lambda](#)
- [Images déconseillées CodeBuild](#)
- [Runtimes disponibles](#)
- [Versions d'environnement d'exécution](#)

Obtenir la liste des images Docker actuelles

CodeBuild met fréquemment à jour la liste des images Docker pour ajouter les dernières images et déprécier les anciennes images. Pour obtenir la liste la plus récente, effectuez l'une des actions suivantes :

- Dans la CodeBuild console, dans l'assistant de création de projet ou sur la page Modifier le projet de génération, pour Image d'environnement, choisissez Image gérée. Effectuez vos sélections dans les listes déroulantes Système d'exploitation, Version d'exécution et Version. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).
- Pour le AWS CLI, exécutez la `list-curated-environment-images` commande :

```
aws codebuild list-curated-environment-images
```

- Pour le AWS SDKs, appelez l'`ListCuratedEnvironmentImages` opération correspondant à votre langage de programmation cible. Pour de plus amples informations, veuillez consulter [AWS SDKs et référence des outils](#).

EC2 calculer des images

AWS CodeBuild prend en charge les images Docker suivantes qui sont disponibles pour le EC2 calcul. CodeBuild

Note

L'image de base de la plateforme Windows Server Core 2019 n'est disponible que dans les régions suivantes :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Oregon)
- Europe (Irlande)

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto8	al/standard/corretto8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto11	al/standard/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
Ubuntu 20.04	aws/codebuild/standard:5.0	Ubuntu/standard/5.0
Ubuntu 22.04	aws/codebuild/standard:6.0	Ubuntu/standard/6.0
Ubuntu 22.04	aws/codebuild/standard:7.0	Ubuntu/standard/7.0
Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A
Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A
Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A
Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/A

Note

Le 22 novembre 2024, les alias des images d'exécution standard basées sur Linux ont été mis à jour de `amazonlinux2` à `amazonlinux`. Aucune mise à jour manuelle n'est requise car les alias précédents sont toujours valides.

Images de calcul Lambda

AWS CodeBuild prend en charge les images Docker suivantes qui sont disponibles pour le AWS Lambda calcul. CodeBuild

⚠ Important

L'on-failureattribut n'est pas pris en charge lors de l'utilisation du calcul Lambda ou de la capacité réservée. Cet attribut ne fonctionne qu'avec les images de EC2 calcul fournies par CodeBuild.

Architecture d'aarch64

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6	lambda/aarch64/dotnet6-6
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8	lambda/aarch64/dotnet8-8
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21	lambda/aarch64/goal-1,21
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.24	lambda/aarch64/goal-1,24
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11	al-lambda/aarch64/corretto 11
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17	al-lambda/aarch64/corretto 17

Plateforme	Identificateur de l'image	Définition
	<code>mbda-standard:corretto17</code>	
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21</code>	lambda/aarch64/correttoal-21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18</code>	al-lambda/aarch64/nodejs 18
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20</code>	al-lambda/aarch64/nodejs 20
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs22</code>	al-lambda/aarch64/nodejs 22
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11</code>	lambda/aarch64/pythonal-3,11
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12</code>	lambda/aarch64/pythonal-3,12

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.13	lambda/aarch64/pythonal-3,13
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2	al-lambda/aarch64/ruby 3.2
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.4	al-lambda/aarch64/ruby 3,4

Architecture d'**x86_64**

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6	lambda/x86_64/dotnetal-6
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8	lambda/x86_64/dotnetal-8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21	lambda/x86_64/goal-1,21

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.24	lambda/x86_64/goal-1,24
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11	al-lambda/x86_64/corretto 11
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17	al-lambda/x86_64/corretto 17
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21	lambda/x86_64/correttoal-21
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18	al-lambda/x86_64/nodejs 18
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20	al-lambda/x86_64/nodejs 20
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs22	al-lambda/x86_64/nodejs 22

Plateforme	Identificateur de l'image	Définition
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11	lambda/x86_64/python3.11
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12	lambda/x86_64/python3.12
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.13	lambda/x86_64/python3.13
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2	al-lambda/x86_64/ruby 3.2
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.4	al-lambda/x86_64/ruby 3.4

Images déconseillées CodeBuild

Une image obsolète est une image qui n'est plus mise en cache ni mise à jour par CodeBuild. Une image obsolète ne reçoit plus de mises à jour de version mineures ni de mises à jour de version de correctif, et comme elles ne sont plus mises à jour, leur utilisation peut ne pas être sécurisée. Si votre CodeBuild projet est configuré pour utiliser une ancienne version d'image, le processus de provisionnement téléchargera cette image docker et l'utilisera pour créer l'environnement d'exécution conteneurisé, ce qui peut augmenter la durée de provisionnement et la durée globale de construction.

CodeBuild a rendu obsolètes les images Docker suivantes. Vous pouvez toujours utiliser ces images, mais elles ne seront pas mises en cache sur l'hôte de compilation et augmenteront les délais de provisionnement.

Plateforme	Identificateur de l'image	Définition	Date d'obsolescence
Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:3.0	al2/standard/3.0	9 mai 2023
Ubuntu 18.04	aws/codebuild/standard:4.0	ubuntu/standard/4.0	31 mars 2023
Amazon Linux 2	aws/codebuild/amazonlinux2-aarch64-standard:1.0	al2/aarch64/standard/1.0	31 mars 2023
Ubuntu 18.04	aws/codebuild/standard:3.0	ubuntu/standard/3.0	30 juin 2022
Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:2.0	al2/standard/2.0	30 juin 2022

Rubriques

- [Runtimes disponibles](#)
- [Versions d'environnement d'exécution](#)

Runtimes disponibles

Vous pouvez spécifier un ou plusieurs runtimes dans la section `runtime-versions` de votre fichier `buildspec`. Si votre runtime dépend d'un autre runtime, vous pouvez également spécifier son runtime

dépendant dans le fichier buildspec. Si vous ne spécifiez aucun environnement d'exécution dans le fichier buildspec, CodeBuild choisit les environnements d'exécution par défaut disponibles dans l'image que vous utilisez. Si vous spécifiez un ou plusieurs environnements d'exécution, utilisez uniquement CodeBuild ces environnements d'exécution. Si aucun environnement d'exécution dépendant n'est spécifié, CodeBuild tente de le choisir pour vous. Pour de plus amples informations, veuillez consulter [Specify runtime versions in the buildspec file](#).

Rubriques

- [Temps d'exécution des images Linux](#)
- [Temps d'exécution des images macOS](#)
- [Temps d'exécution des images Windows](#)

Temps d'exécution des images Linux

Le tableau suivant contient les environnements d'exécution disponibles et les images Linux standard qui les prennent en charge.

Runtimes des plateformes Ubuntu et Amazon Linux

Nom de l'exécution	Version	Images
dotnet	6.0	Amazon Linux 2 x86_64 Norme Lambda : dotnet6
		Norme AArch64 Lambda Amazon Linux 2 : dotnet6
		Amazon Linux 2023 x86_64 standard : 5.0
		Amazon Linux 2023 AArch64 standard : 3.0
		Norme Ubuntu : 7.0
	8.0	Amazon Linux 2023 x86_64 standard : 5.0

Nom de l'exécution	Version	Images
golang		Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	1,20	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	1,21	Amazon Linux 2 x86_64 norme Lambda : go1.21 Norme AArch64 Lambda Amazon Linux 2 : go1.21 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	1,22	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	1,23	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	1,24	Amazon Linux 2023 x86_64 Norme Lambda : go1.24 Norme AArch64 Lambda d'Amazon Linux 2023 : go1.24 Amazon Linux 2023 x86_64 standard : 5.0 Norme Ubuntu : 7.0
java	corretto8	Amazon Linux 2 x86_64 standard : corretto8 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	corretto11	Amazon Linux 2 x86_64 standard : corretto11 Amazon Linux 2 x86_64 Norme Lambda : corretto11 Amazon Linux 2023 x86_64 standard : 5.0 Norme AArch64 Lambda d'Amazon Linux 2 : corretto11 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	corretto17	Amazon Linux 2 x86_64 Norme Lambda : corretto17 Norme AArch64 Lambda d'Amazon Linux 2 : corretto17 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	corretto21	Amazon Linux 2 x86_64 Norme Lambda : corretto21 Norme AArch64 Lambda d'Amazon Linux 2 : corretto21 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	corretto25	Amazon Linux 2023 AArch64 standard : 3.0
nodejs	18	Amazon Linux 2 x86_64 Norme Lambda : nodejs18 Norme AArch64 Lambda Amazon Linux 2 : nodejs18 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	20	Amazon Linux 2 x86_64 Norme Lambda : nodejs20 Norme AArch64 Lambda Amazon Linux 2 : nodejs20 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	22	Amazon Linux 2023 x86_64 Norme Lambda : nodejs22 Norme AArch64 Lambda d'Amazon Linux 2023 : nodejs22 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	24	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
php	8.1	Amazon Linux 2023 AArch64 standard : 3.0
	8.2	Amazon Linux 2023 x86_64 standard : 5.0
		Amazon Linux 2023 AArch64 standard : 3.0
		Norme Ubuntu : 7.0
	8,3	Amazon Linux 2023 x86_64 standard : 5.0
		Amazon Linux 2023 AArch64 standard : 3.0
		Norme Ubuntu : 7.0
	8,4	Amazon Linux 2023 AArch64 standard : 3.0
	8,5	
		Amazon Linux 2023 AArch64 standard : 3.0
python	3.9	Amazon Linux 2023 x86_64 standard : 5.0
		Amazon Linux 2023 AArch64 standard : 3.0
		Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	3,10	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	3,11	Amazon Linux 2 x86_64 Norme Lambda : python3.11 Norme AArch64 Lambda Amazon Linux 2 : python3.11 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	3,12	Amazon Linux 2 x86_64 Norme Lambda : python3.12 Norme AArch64 Lambda Amazon Linux 2 : python3.12 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	3.13	Amazon Linux 2023 x86_64 Norme Lambda : python3.13 Norme AArch64 Lambda d'Amazon Linux 2023 : python3.13 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	3,14	Amazon Linux 2023 AArch64 standard : 3.0
ruby	3.1	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Nom de l'exécution	Version	Images
	3.2	Amazon Linux 2 x86_64 Norme Lambda : ruby3.2 Norme AArch64 Lambda Amazon Linux 2 : ruby3.2 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	3.3	Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0
	3.4	Amazon Linux 2023 x86_64 Norme Lambda : ruby3.4 Norme AArch64 Lambda d'Amazon Linux 2023 : ruby3.4 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0

Temps d'exécution des images macOS

Important

Les images CodeBuild sélectionnées pour les versions Mac contiennent des macOS et Xcode préinstallés. En utilisant le logiciel Xcode, vous reconnaissez, comprenez et consentez à l'accord entre [Xcode et Apple SDKs](#). Si vous n'acceptez pas les termes et conditions du contrat, n'utilisez pas le logiciel Xcode. Fournissez plutôt vos propres Amazon Machine Images (AMI). Pour de plus amples informations, consultez [Comment configurer une flotte de macOS à capacité réservée ?](#).

Le tableau suivant répertorie les environnements d'exécution disponibles pris en charge par macOS.

Runtimes de la plateforme macOS

Nom de l'exécution	Version	Images	Informations complémentaires
bash	3,2,57	macos-arm-base:15	
		macos-arm-base:26	
s'accrocha	17,0,0	macos-arm-base:15	
		macos-arm-base:26	
SDK Dotnet	8,0,416	macos-arm-base:15	
	8,0,417	macos-arm-base:26	
	10,0,101	macos-arm-base:15	
	10,0,102	macos-arm-base:26	
gcc	11,5,0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gcc-11alias
	12.4.0	macos-arm-base:15	Disponible en utilisant l'gcc-12alias

Nom de l'exécution	Version	Images	Informations complémentaires
		macos-arm-base:26	
	13.4.0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gcc-13alias
	14.3.0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gcc-14alias
gnu	11,5,0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gfortran-11 alias
	12.4.0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gfortran-12 alias
	13.4.0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gfortran-13 alias
	14.3.0	macos-arm-base:15 macos-arm-base:26	Disponible en utilisant l'gfortran-14 alias
golang	1,24,11	macos-arm-base:15 macos-arm-base:26	
	1.25.4	macos-arm-base:15 macos-arm-base:26	
java	Corretto 8	macos-arm-base:15 macos-arm-base:26	

Nom de l'exécution	Version	Images	Informations complémentaires
	Corretto 11	macos-arm-base:15 macos-arm-base:26	
	Corretto 17	macos-arm-base:15 macos-arm-base:26	
	Corretto 21	macos-arm-base:15 macos-arm-base:26	
	Corretto 25	macos-arm-base:15 macos-arm-base:26	
kotlin	2.2.21	macos-arm-base:15	
	2.3.0	macos-arm-base:26	
mono	6.14.1	macos-arm-base:15 macos-arm-base:26	
nodejs	20,19,6	macos-arm-base:15 macos-arm-base:26	
	22.21.1	macos-arm-base:15 macos-arm-base:26	
	24,11.1	macos-arm-base:15 macos-arm-base:26	
perl	5,34.1	macos-arm-base:15 macos-arm-base:26	

Nom de l'exécution	Version	Images	Informations complémentaires
php	8,2,29	macos-arm-base:15	
	8,2,30	macos-arm-base:26	
	8,3,28	macos-arm-base:15	
	8,3,30	macos-arm-base:26	
	8,4,15	macos-arm-base:15	
	8,4,17	macos-arm-base:26	
	8.5.0	macos-arm-base:15	
	8.5.2	macos-arm-base:26	
python	3,1,19	macos-arm-base:15	
		macos-arm-base:26	
	3,1,114	macos-arm-base:15	
		macos-arm-base:26	
	3,12,12	macos-arm-base:15	
		macos-arm-base:26	
ruby	3,13,10	macos-arm-base:15	
		macos-arm-base:26	
	3.14.1	macos-arm-base:15	
		macos-arm-base:26	
ruby	3.2.9	macos-arm-base:15	
		macos-arm-base:26	

Nom de l'exécution	Version	Images	Informations complémentaires
	3.3.10	macos-arm-base:15	
		macos-arm-base:26	
	3.4.7	macos-arm-base:15	
		macos-arm-base:26	
rust	1,91,1	macos-arm-base:15	
	1,92,0	macos-arm-base:26	
rapide	6.2.1	macos-arm-base:15	
	6.2.3	macos-arm-base:26	
Xcode	26.1.1	macos-arm-base:15	
	26,2	macos-arm-base:26	

Temps d'exécution des images Windows

L'image de base de Windows Server Core 2019 contient les environnements d'exécution suivants.

Runtimes de la plateforme Windows

Nom de l'exécution	Windows Server Core 2019 standard : versions 1.0	Windows Server Core 2019 standard : versions 2.0	Windows Server Core 2019 standard : versions 3.0
dotnet	3.1	3.1	8.0
	5.0	6.0	
		7.0	
SDK Dotnet	3.1	3.1	8.0

Nom de l'exécution	Windows Server Core 2019 standard : versions 1.0	Windows Server Core 2019 standard : versions 2.0	Windows Server Core 2019 standard : versions 3.0
	5.0	6.0 7.0	
golang	1.14	1,18	1,21 1,22 1,23
gradle	6.7	7.6	8,12
java	Corretto 11	Corretto 11 Corretto 17	Corretto 8 Corretto 11 Corretto 17 Corretto 21
maven	3.6	3.8	3.9
nodejs	14,15	16,19	20,18 22,13
php	7.4	8.1	8,3 8,4
powershell	7.1	7.2	7.4
python	3.8	3,10	3,10 3,11 3,12 3.13

Nom de l'exécution	Windows Server Core 2019 standard : versions 1.0	Windows Server Core 2019 standard : versions 2.0	Windows Server Core 2019 standard : versions 3.0
ruby	2.7	3.1	3.2
			3.3
			3.4

Versions d'environnement d'exécution

Lorsque vous spécifiez un environnement d'exécution dans la section [runtime-versions](#) de votre fichier buildspec, vous pouvez spécifier une version spécifique, une version majeure spécifique et la dernière version mineure, ou la dernière version. Le tableau suivant répertorie les environnements d'exécution disponibles et décrit comment les spécifier. Les versions d'exécution ne sont pas toutes disponibles sur toutes les images. La sélection de la version d'exécution n'est pas non plus prise en charge pour les images personnalisées. Pour de plus amples informations, veuillez consulter [Runtimes disponibles](#). Si vous souhaitez installer et utiliser une version d'exécution personnalisée au lieu des versions d'exécution préinstallées, consultez [Versions d'exécution personnalisées](#).

Versions d'exécution des plateformes Ubuntu et Amazon Linux 2

Nom de l'exécution	Version	Version spécifique	Version spécifique majeure et dernière version mineure	Dernière version
android	28	android: 28	android: 28.x	android: latest
	29	android: 29	android: 29.x	
dotnet	3.1	dotnet: 3.1	dotnet: 3.x	dotnet: latest
	5.0	dotnet: 5.0	dotnet: 5.x	
	6.0	dotnet: 6.0	dotnet: 6.x	
	8.0	dotnet: 8.0	dotnet: 8.x	

Nom de l'exécution	Version	Version spécifique	Version spécifique majeure et dernière version mineure	Dernière version
golang	1.12	golang: 1.12	golang: 1.x	golang: latest
	1.13	golang: 1.13		
	1.14	golang: 1.14		
	1.15	golang: 1.15		
	1.16	golang: 1.16		
	1.18	golang: 1.18		
	1.20	golang: 1.20		
	1.21	golang: 1.21		
	1.22	golang: 1.22		
	1.23	golang: 1.23		
	1.24	golang: 1.24		
java	corretto8	java: corretto	java: corretto .x	java: latest
	corretto11	java: corretto 1	java: corretto 1.x	
	corretto17	java: corretto 7	java: corretto 7.x	
	corretto21	java: corretto 1	java: corretto 1.x	
nodejs	10	nodejs: 10	nodejs: 10.x	nodejs: latest

Nom de l'exécution	Version	Version spécifique	Version spécifique majeure et dernière version mineure	Dernière version
	12	nodejs: 12	nodejs: 12.x	
	14	nodejs: 14	nodejs: 14.x	
	16	nodejs: 16	nodejs: 16.x	
	18	nodejs: 18	nodejs: 18.x	
	20	nodejs: 20	nodejs: 20.x	
	22	nodejs: 22	nodejs: 22.x	
php	7.3	php: 7.3	php: 7.x	php: latest
	7.4	php: 7.4		
	8.0	php: 8.0	php: 8.x	
	8.1	php: 8.1		
	8.2	php: 8.2		
	8.3	php: 8.3		
python	3.7	python: 3.7	python: 3.x	python: latest
	3.8	python: 3.8		
	3.9	python: 3.9		
	3.10	python: 3.10		
	3.11	python: 3.11		
	3.12	python: 3.12		

Nom de l'exécution	Version	Version spécifique	Version spécifique majeure et dernière version mineure	Dernière version
	3.13	python: 3.13		
ruby	2.6	ruby: 2.6	ruby: 2.x	ruby: latest
	2.7	ruby: 2.7		
	3.1	ruby: 3.1	ruby: 3.x	
	3.2	ruby: 3.2		
	3.3	ruby: 3.3		
	3.4	ruby: 3.4		

Vous pouvez utiliser une spécification de construction pour installer d'autres composants (par exemple AWS CLI, Apache Maven, Apache Ant, Mocha ou similaire) pendant la phase de `install` construction. RSpec Pour de plus amples informations, veuillez consulter [Exemple de fichier buildspec](#).

Versions d'exécution personnalisées

Au lieu d'utiliser les versions d'exécution préinstallées dans des images CodeBuild gérées, vous pouvez installer et utiliser les versions personnalisées de votre choix. Le tableau suivant répertorie les environnements d'exécution personnalisés disponibles et indique comment les spécifier.

Note

La sélection personnalisée de la version d'exécution n'est prise en charge que pour les images Ubuntu et Amazon Linux.

Versions d'exécution personnalisées

Nom de l'exécution	Syntaxe	Exemple
dotnet	<i><major>.<minor>.<patch></i>	5.0.408
golang	<i><major>.<minor></i>	1.19
	<i><major>.<minor>.<patch></i>	1.19.1
java	corretto <i><major></i>	corretto15
nodejs	<i><major></i>	14
	<i><major>.<minor></i>	14.21
	<i><major>.<minor>.<patch></i>	14.21.3
php	<i><major>.<minor>.<patch></i>	8.0.30
python	<i><major></i>	3
	<i><major>.<minor></i>	3.7
	<i><major>.<minor>.<patch></i>	3.7.16
ruby	<i><major>.<minor>.<patch></i>	3.0.6

Exemple de buildspec d'exécution personnalisé

Voici un exemple de buildspec qui spécifie des versions d'exécution personnalisées.

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto15
      php: 8.0.30
      ruby: 3.0.6
      golang: 1.19
      python: 3.7
      nodejs: 14
```

`dotnet: 5.0.408`

Modes et types de calcul de l'environnement de création

Dans CodeBuild, vous pouvez spécifier l'image de l'environnement de calcul et d'exécution CodeBuild à utiliser pour exécuter vos builds. Le calcul fait référence au moteur informatique (le processeur, la mémoire et le système d'exploitation) qui est géré et maintenu par CodeBuild. Une image d'environnement d'exécution est une image de conteneur qui s'exécute au-dessus de la plateforme de calcul que vous avez choisie et inclut des outils supplémentaires dont votre build pourrait avoir besoin, tels que le AWS CLI.

Rubriques

- [À propos du calcul](#)
- [À propos des types d'environnement de capacité réservée](#)
- [À propos des types d'environnements à la demande](#)

À propos du calcul

CodeBuild offre EC2 et modes AWS Lambda de calcul. EC2 offre une flexibilité optimisée lors de votre construction et AWS Lambda des vitesses de démarrage optimisées. AWS Lambda prend en charge des builds plus rapides grâce à une latence de démarrage plus faible. AWS Lambda évolue également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Pour de plus amples informations, veuillez consulter [Exécuter des builds sur AWS Lambda ordinateur](#).

Avec le mode EC2 informatique, vous pouvez exécuter vos builds avec des flottes de capacité à la demande ou réservées. Pour les flottes à la demande, vous pouvez sélectionner des types de calcul prédéfinis tels BUILD_GENERAL1_SMALL que ou. BUILD_GENERAL1_LARGE Pour de plus amples informations, veuillez consulter [À propos des types d'environnements à la demande](#). Pour les flottes de capacité réservées, vous pouvez sélectionner vos configurations de calcul, notamment le vCPU, la mémoire et l'espace disque. Après avoir spécifié les configurations, CodeBuild vous choisirez un type de calcul pris en charge qui correspond à vos besoins. Pour de plus amples informations, veuillez consulter [À propos des types d'environnement de capacité réservée](#).

À propos des types d'environnement de capacité réservée

AWS CodeBuild fournit les types d'environnement Linux x86, Arm, GPU, Windows et macOS pour les flottes de capacité réservée. Le tableau suivant indique le type de machine, la mémoire, le v et l'espace disque disponibles CPUs, triés par région :

US East (N. Virginia)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	64	128 Gio	824 Go	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 Go SSD	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 Gio	885 GO (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	64	256 Gio	185 GO (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
GPU Linux	96	384 Go	3785 GO (SSD)	NVME	reserved.gpu.96cpu.384gib.nvme
macOS	8	24 GiO	128 Go	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiO	256 Go	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

US East (Ohio)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	48	96 GiB	824 Go SSD	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GO (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 Gio	885 GO (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiO	128 Go	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiO	256 Go	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

US West (Oregon)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 Go SSD	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 Gio	885 GO (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
GPU Linux	64	256 Gio	185 GO (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
macOS	8	24 GiO	128 Go	GENERAL	reserved.arm.m2.8cpu.24gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
macOS	12	32 GiO	256 Go	GENERAL	reserved. arm.m2.12 cpu.32gib
Windows	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	72	144 GiB	824 GO	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved.x86-64.96cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Tokyo)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Mumbai)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Singapore)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 Go SSD	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved.x86-64.72cpu.144gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	96	192 Go	824 GO	GENERAL	reserved.x86-64.96cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Sydney)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gi b
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gi b

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gi b.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvm e
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved. gpu.8cpu. 32gib.nvm e
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved. gpu.16cpu .64gib.nv me

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiO	128 Go	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiO	256 Go	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Europe (Frankfurt)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiO	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 Gio	885 GO (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiO	128 Go	GENERAL	reserved.arm.m2.8cpu.24gib
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gi b
Windows	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gi b
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Europe (Ireland)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 Gio	824 GO	GENERAL	reserved.arm.64cpu.128gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 Go	824 GO	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 Go SSD	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gib .nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiO	235 GO (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiO	435 GO (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 Go	585 GO (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 Gio	885 GO (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
GPU Linux	48	192 Go	3785 GO (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	72	144 GiB	824 GO	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 Go	824 GO	GENERAL	reserved.x86-64.96cpu.192gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved.x86-64.8cpu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

South America (São Paulo)

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
ARM	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiO	256 Go	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 Go	256 Go	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 Go	GENERAL	reserved.arm.48cpu.96gib
BRAS EC2	2	4 GiB	64 Go	GENERAL	reserved.arm.2cpu.4gib
BRAS EC2	4	8 GiO	128 Go	GENERAL	reserved.arm.4cpu.8gib
BRAS EC2	8	16 GiO	128 Go	GENERAL	reserved.arm.8cpu.16gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	2	4 GiB	64 Go	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 Go	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 Go	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 Go	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 Go	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 Go	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GO	GENERAL	reserved.x86-64.72cpu.144gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Linux	72	144 GiB	824 Go SSD	NVME	reserved. x86-64.72 cpu.144gi b.nvme
Linux EC2	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Windows	2	4 GiB	64 Go	GENERAL	reserved. x86-64.2c pu.4gib
Windows	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiO	256 Go	GENERAL	reserved. x86-64.16 cpu.32gib

Type d'environnement	v CPUs	Mémoire	Espace disque	Type de machine	Type d'instance de calcul
Windows	36	72 GiB	256 Go	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 Go	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GO	GENERAL	reserved. x86-64.72 cpu.144gib
Fenêtres EC2	4	8 GiO	128 Go	GENERAL	reserved. x86-64.4c pu.8gib
Fenêtres EC2	8	16 GiO	128 Go	GENERAL	reserved. x86-64.8c pu.16gib

Pour plus d'informations sur l'identifiant de tarification, voir <https://aws.amazon.com/codebuild/pricing/>.

Pour choisir un type de calcul :

- Dans la CodeBuild console, sur la page de configuration du parc de calcul, choisissez l'une des options parmi v CPUs, Memory et Disk. Pour de plus amples informations, veuillez consulter [Création d'un parc de capacités réservées](#).

- Pour le AWS CLI, exécutez la `update-fleet` commande `create-fleet` or en spécifiant la valeur de `computeType` `toATTRIBUTE_BASED_COMPUTE`. [Pour plus d'informations, voir `create-fleet` ou `update-fleet`.](#)
- Pour le AWS SDKs, appelez l'équivalent de l'UpdateFleetopération `CreateFleet` ou pour votre langage de programmation cible, en spécifiant la valeur de `computeType` `toATTRIBUTE_BASED_COMPUTE`. Pour de plus amples informations, veuillez consulter [AWS SDKs et référence des outils](#).

Note

Pour le AWS CLI et AWS SDKs, vous pouvez toujours utiliser `computeType` des entrées telles que `BUILD_GENERAL1_SMALL`, pour choisir les types de calcul au lieu de `ATTRIBUTE_BASED_COMPUTE`. Pour de plus amples informations, veuillez consulter [À propos des types d'environnements à la demande](#).

Familles d'instances prises en charge

AWS CodeBuild prend en charge les instances suivantes pour les flottes de capacité réservée. :

- Usage général : M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8g | T3 | T3a | T4g
- Optimisées pour le calcul : C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8g
- Mémoire optimisée : R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7i | R8g | U-3tb1 | U-6TB1 | U-6TB1 | U-9b TB1 | U-12TB1 | U-18TB1 | U-24TB1 | U7i-6 TB | U7i-8TB | U7i-12TB | U7in-16TB | U7in-24TB | U7in-32TB | X1e | X2GD | X2IDN | X2ieZN | x8g | z1d
- Stockage optimisé : D3 | D3en | I3 | i3EN | i4G | i4i | i7IE | i8G | i4GN | IS4gen
- Calcul accéléré : DL1 | F1 DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | Gr6 | Inf1 | Inf2 | P3 | P3dn | P4d | P5 | P5e | P5en | Trn1 | Trn1n | Trn2 | VT1
- Calcul à hautes performances : Hpc6a | Hpc6id | Hpc7a | Hpc7g
- Génération précédente : A1

Pour créer un parc de capacités réservées avec un type d'instance spécifique :

- Dans la CodeBuild console, sur la page de configuration du parc de calcul, accédez à la section Configuration de la capacité. En mode de sélection de calcul, sélectionnez Saisie manuelle et dans Type d'instance de calcul, choisissez l'un des types d'instance dans le menu déroulant. Pour de plus amples informations, veuillez consulter [Création d'un parc de capacités réservées](#).
- Pour le AWS CLI, exécutez la `update-fleet` commande `create-fleet` or, en spécifiant la valeur de `computeType` to `CUSTOM_INSTANCE_TYPE` et de `ComputeConfiguration` `instanceType` to pour le type d'instance spécifié. [Pour plus d'informations, voir `create-fleet` ou `update-fleet`](#).
- Pour le AWS SDKs, appelez l'équivalent de l'UpdateFleetopération `CreateFleet` or pour votre langage de programmation cible, en spécifiant la valeur de `computeType` to `CUSTOM_INSTANCE_TYPE` et de `ComputeConfiguration``instanceType`to pour le type d'instance spécifié. Pour de plus amples informations, veuillez consulter [AWS SDKs et référence des outils](#).

À propos des types d'environnements à la demande

AWS CodeBuild fournit des environnements de construction avec la mémoire, le v CPUs et l'espace disque disponibles suivants pour le mode EC2 calcul :

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	v CPUs	Espace disque
ARM Small ¹	BUILD_GENERAL1_SMALL	ARM_CONTAINER ARM_EC2	4 GiB	2	64 Go
ARM Medium ¹	BUILD_GENERAL1_MEDIUM	ARM_CONTAINER ARM_EC2	8 GiO	4	128 Go
ARM Large ¹	BUILD_GENERAL1_LARGE	ARM_CONTAINER ARM_EC2	16 GiO	8	128 Go

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	v CPUs	Espace disque
ARM_EC2					
BRAS XLarge ¹	BUILD_GENERAL1_XLARGE	ARM_CONTAINER	64 Go	32	256 Go
BRAS 2 XLarge ¹	BUILD_GENERAL1_2XLARGE	ARM_CONTAINER	96 GiB	48	824 GO
Linux Small ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER LINUX_EC2	4 GiB	2	64 Go
Linux Medium ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER LINUX_EC2	8 GiO	4	128 Go
Linux Large ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER LINUX_EC2	16 GiO	8	128 Go
Linux XLarge ¹	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	72 GiB	36	256 Go
Linux 2 XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	144 GiB	72	824 Go SSD

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	v CPUs	Espace disque
GPU Linux petit	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GiO	4	235 GO (SSD)
GPU Linux de grande taille	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GiB	32	50 Go
Windows Medium ¹	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER WINDOWS_SERVER_2022_CONTAINER WINDOWS_EC2	8 GiO	4	128 Go

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	v CPUs	Espace disque
Windows Large ¹	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER WINDOWS_SERVER_2022_CONTAINER WINDOWS_EC2	16 GiO	8	128 Go
Windows XLarge ¹	BUILD_GENERAL1_XLARGE	WINDOWS_SERVER_2022_CONTAINER	72 GiB	36	256 Go
Windows 2 XLarge ¹	BUILD_GENERAL1_2XLARGE	WINDOWS_SERVER_2022_CONTAINER	144 GiB	72	824 GO

¹ La dernière version de ce type d'image est mise en cache. Si vous spécifiez une version plus spécifique, CodeBuild provisionne cette version au lieu de la version mise en cache. Cela peut entraîner des temps de génération plus longs. Par exemple, pour tirer profit de la mise en cache, spécifiez `aws/codebuild/amazonlinux-x86_64-standard:5.0` à la place d'une version plus précise, telle que `aws/codebuild/amazonlinux-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild fournit des environnements de construction dotés de la mémoire et de l'espace disque disponibles suivants pour le mode AWS Lambda calcul :

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	Espace disque
ARM Lambda 1 Go	BUILD_LAMBDA_1GB	ARM_LAMBDA_CONTAINER	1 Gio	10 Go
ARM Lambda 2 Go	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GiB	10 Go
ARM Lambda 4 Go	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GiB	10 Go
ARM Lambda 8 Go	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GiB	10 Go
ARM Lambda 10 Go	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 Gio	10 Go
Linux Lambda 1 Go	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	1 Gio	10 Go
Linux Lambda 2 Go	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GiB	10 Go
Linux Lambda 4 Go	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GiB	10 Go

Type de calcul	Valeur ComputeType de l'environnement	Valeur du type d'environnement	Mémoire	Espace disque
Linux Lambda 8 Go	BUILD_LAMBDA_8GB	LINUX_LAMBDA_CONTAINER	8 GiO	10 Go
Linux Lambda 10 Go	BUILD_LAMBDA_10GB	LINUX_LAMBDA_CONTAINER	10 Gio	10 Go

Lorsque vous utilisez d'autres types d'environnement, il est recommandé d'utiliser une image mise en cache pour réduire les temps de création.

L'espace disque répertorié pour chaque environnement de construction n'est disponible que dans le répertoire spécifié par la variable d'environnement CODEBUILD_SRC_DIR.

Pour choisir un type de calcul :

- Dans la CodeBuild console, dans l'assistant de création de projet ou sur la page Modifier le projet de génération, dans Environnement, développez Configuration supplémentaire, puis choisissez l'une des options dans Type de calcul. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).
- Pour le AWS CLI, exécutez la update-project commande create-project ou en spécifiant la computeType valeur de l'environnementobjet. Pour plus d'informations, consultez [Création d'un projet de génération \(AWS CLI\)](#) ou [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).
- Pour le AWS SDKs, appelez l'équivalent de l'UpdateProjectopération CreateProject ou pour votre langage de programmation cible, en spécifiant l'équivalent de la computeType valeur de l'environnementobjet. Pour de plus amples informations, veuillez consulter [AWS SDKs et référence des outils](#).

Certains types d'environnement et de calcul présentent des limites de disponibilité par région :

- Le type de calcul Linux GPU Small (LINUX_GPU_CONTAINER) n'est disponible que dans les régions suivantes :

- USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie Pacifique (Tokyo)
 - Canada (Centre)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
- Le type de calcul Linux GPU Large (LINUX_GPU_CONTAINER) n'est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Séoul)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Le type de calcul n'BUILD_GENERAL1_2XLARGE est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Californie du Nord)
 - US West (Oregon)
 - Asie-Pacifique (Hyderabad)
 - Asie-Pacifique (Hong Kong)
 - Asie-Pacifique (Jakarta)
 - Asie-Pacifique (Melbourne)

- Asie-Pacifique (Mumbai)
 - Asie-Pacifique (Séoul)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Europe (Paris)
 - Europe (Espagne)
 - Europe (Stockholm)
 - Europe (Zurich)
 - Israël (Tel Aviv)
 - Moyen-Orient (Bahreïn)
 - Moyen-Orient (EAU)
 - Amérique du Sud (São Paulo)
- Le type d'environnement n'ARM_CONTAINER est disponible que dans les régions suivantes :
- USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Californie du Nord)
 - US West (Oregon)
 - Asie-Pacifique (Hong Kong)
 - Asie-Pacifique (Jakarta)
 - Asie-Pacifique (Hyderabad)
 - Asie-Pacifique (Mumbai)
 - **Asie-Pacifique (Osaka)**

- Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Europe (Milan)
 - Europe (Paris)
 - Europe (Espagne)
 - Europe (Stockholm)
 - Israël (Tel Aviv)
 - Moyen-Orient (Bahreïn)
 - Moyen-Orient (EAU)
 - Amérique du Sud (São Paulo)
- Le type d'environnement `n'WINDOWS_SERVER_2022_CONTAINER` est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Europe (Francfort)
 - Europe (Irlande)
 - South America (São Paulo)
 - Le type d'environnement `LINUX_EC2` (`BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) n'est disponible que dans les régions suivantes :
 - USA Est (Ohio)

- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Melbourne)
- Europe (Zurich)
- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Chine (Beijing)
- China (Ningxia)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Espagne)
- Europe (Stockholm)
- Israël (Tel Aviv)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (EAU)
- **Amérique du Sud (São Paulo)**
- AWS GovCloud (US-Ouest)

- AWS GovCloud (USA Est)
- Le type d'environnement ARM_EC2 (BUILD_GENERAL1_SMALL,BUILD_GENERAL1_MEDIUM,BUILD_GENERAL1_LARGE) n'est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Californie du Nord)
 - US West (Oregon)
 - Asie-Pacifique (Hong Kong)
 - Asie-Pacifique (Jakarta)
 - Europe (Zurich)
 - Asie-Pacifique (Hyderabad)
 - Asie-Pacifique (Mumbai)
 - Asie-Pacifique (Osaka)
 - Asia Pacific (Seoul)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Europe (Milan)
 - Europe (Paris)
 - Europe (Espagne)
 - Europe (Stockholm)
 - Israël (Tel Aviv)
 - Moyen-Orient (Bahreïn)
 - Amérique du Sud (São Paulo)

- AWS GovCloud (US-Ouest)
- AWS GovCloud (USA Est)
- Le type d'environnement `WINDOWS_EC2` (`BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) n'est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Europe (Francfort)
 - Europe (Irlande)
 - South America (São Paulo)
- Le mode de calcul AWS Lambda (`ARM_LAMBDA_CONTAINER`, `LINUX_LAMBDA_CONTAINER`) n'est disponible que dans les régions suivantes :
 - USA Est (Virginie du Nord)
 - USA Est (Ohio)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Mumbai)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Europe (Francfort)
 - Europe (Irlande)
 - South America (São Paulo)
- Le mode de calcul `n1.M4` est disponible que dans les régions suivantes :
 - USA Est (Virginie du Nord)
 - USA Est (Ohio)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Sydney)

Pour le type de calcul BUILD_GENERAL1_2XLARGE, les images Docker jusqu'à 100 Go non compressées sont prises en charge.

 Note

Pour les images d'environnement de construction personnalisées, CodeBuild prend en charge les images Docker jusqu'à 50 Go non compressées sous Linux et Windows, quel que soit le type de calcul. Pour vérifier la taille de votre image de génération, utilisez Docker pour exécuter la commande `docker images` *REPOSITORY:TAG*.

Vous pouvez utiliser Amazon EFS pour accéder à davantage d'espace dans votre conteneur de build. Pour de plus amples informations, veuillez consulter [Exemple d'Amazon Elastic File System pour AWS CodeBuild](#). Si vous souhaitez manipuler l'espace disque du conteneur pendant une génération, la version de la génération doit s'exécuter en mode privilégié.

 Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux](#) Capabilities sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

Shells et commandes dans les environnements de génération

Vous fournissez un ensemble de commandes AWS CodeBuild à exécuter dans un environnement de génération pendant le cycle de vie d'une version (par exemple, en installant les dépendances de version et en testant et compilant votre code source). Vous pouvez spécifier les commandes suivantes de plusieurs façons :

- Créez un fichier de spécification de génération et incluez-le avec votre code source. Dans ce fichier, spécifiez les commandes que vous souhaitez exécuter dans chaque phase du cycle de vie de génération. Pour plus d'informations, consultez le [Référence de spécification de construction pour CodeBuild](#).
- Utilisez la CodeBuild console pour créer un projet de génération. Dans Insérer des commandes de génération, pour Build commands (Commandes de génération), saisissez les commandes

que vous souhaitez exécuter lors de la phase build. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).

- Utilisez la CodeBuild console pour modifier les paramètres d'un projet de génération. Dans Insérer des commandes de génération, pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase build. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).
- Utilisez le AWS CLI ou AWS SDKs pour créer un projet de construction ou modifier les paramètres d'un projet de construction. Faites référence au code source qui contient un fichier de spécification de génération avec vos commandes, ou spécifiez une chaîne unique qui inclut le contenu d'un fichier de spécification de génération équivalent. Pour plus d'informations, consultez [Création d'un projet de génération](#) ou [Modifier les paramètres du projet de construction](#).
- Utilisez le AWS CLI ou AWS SDKs pour démarrer une compilation, en spécifiant un fichier buildspec ou une chaîne unique incluant le contenu d'un fichier buildspec équivalent. Pour de plus amples informations, consultez la description de la valeur buildspecOverride dans [Exécuter les builds manuellement](#).

Vous pouvez spécifier n'importe quelle commande Shell Command Language (sh). Dans la version 0.1 de buildspec, CodeBuild exécute chaque commande Shell dans une instance distincte de l'environnement de construction. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par conséquent, par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes (par exemple, pour le changement de répertoire ou la définition de variables d'environnement). Pour contourner ce problème, nous vous recommandons d'utiliser la version 0.2, qui permet de résoudre ce problème. Si vous devez utiliser la version 0.1, nous vous recommandons les approches suivantes :

- Inclure dans votre code source un script shell qui contient les commandes que vous souhaitez exécuter dans une instance unique du shell par défaut. Par exemple, vous pouvez inclure dans votre code source un fichier nommé `my-script.sh` qui contient des commandes comme `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd`. Ensuite, dans votre fichier de spécification de génération, spécifiez la commande `./my-script.sh`.
- Dans votre fichier de spécification de génération, ou pour le paramètre Build commands (Commandes de génération) pour la phase build uniquement, entrez une commande unique incluant toutes les commandes que vous souhaitez exécuter dans une instance unique de l'interpréteur de commandes par défaut (par exemple, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

En CodeBuild cas d'erreur, il peut être plus difficile de résoudre l'erreur que d'exécuter une seule commande dans sa propre instance du shell par défaut.

Les commandes exécutées dans une image Windows Server Core utilisent le PowerShell shell.

Variables d'environnement dans les environnements de génération

AWS CodeBuild fournit plusieurs variables d'environnement que vous pouvez utiliser dans vos commandes de compilation :

AWS_DEFAULT_RÉGION

La AWS région dans laquelle le build est en cours d'exécution (par exemple, `us-east-1`). Cette variable d'environnement est utilisée principalement par l' AWS CLI.

AWS_REGION

La AWS région dans laquelle le build est en cours d'exécution (par exemple, `us-east-1`). Cette variable d'environnement est utilisée principalement par l' AWS SDKs.

CODEBUILD_BATCH_BUILD_IDENTIFIER

L'identifiant de la construction dans une génération par lots. Ceci est spécifié dans les spécifications de construction par lots. Pour de plus amples informations, veuillez consulter [the section called “Référence des spécifications de construction par lots”](#).

CODEBUILD_BUILD_ARN

Le nom de ressource Amazon (ARN) de la version (par exemple, `arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

L' CodeBuild ID de la version (par exemple, `codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

L'identifiant de l'image de CodeBuild construction (par exemple, `aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

Numéro de version actuel du projet.

CODEBUILD_BUILD_SUCCEEDING

Si la version actuelle est réussie. Défini sur 0 si la génération échoue, ou sur 1 si la génération aboutit.

CODEBUILD_INITIATOR

Entité à l'origine de la création. Si vous avez CodePipeline démarré la construction, il s'agit du nom du pipeline (par exemple,codepipeline/my-demo-pipeline). Si un utilisateur a lancé le build, il s'agit du nom de l'utilisateur (par exemple,MyUserName). Si le plugin Jenkins pour CodeBuild a démarré la construction, il s'agit de la chaîneCodeBuild-Jenkins-Plugin.

CODEBUILD_KMS_KEY_ID

Identifiant de la AWS KMS clé CodeBuild utilisée pour chiffrer l'artefact de sortie de construction (par exemple, arn:aws:kms:*region-ID*:*account-ID*:key/*key-ID* ou alias/*key-alias*).

CODEBUILD_PROJECT_ARN

Le nom de ressource Amazon (ARN) du projet (par exemple,arn:aws:codebuild:*region-ID*:*account-ID*:project/*project-name*).

CODEBUILD_PUBLIC_BUILD_URL

L'URL des résultats de cette version sur le site Web public des versions. Cette variable n'est définie que si les versions publiques sont activées dans le projet de construction. Pour de plus amples informations, veuillez consulter [Obtenir un projet de construction public URLs](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

L'identifiant de version du code source d'une version. Le contenu dépend du référentiel de code source :

CodeCommit GitHub, GitHub Enterprise Server et Bitbucket

Cette variable contient l'ID de validation.

CodePipeline

Cette variable contient la révision source fournie par CodePipeline.

S'il n' CodePipeline est pas possible de résoudre la révision de la source, par exemple lorsque la source est un compartiment Amazon S3 pour lequel le versionnement n'est pas activé, cette variable d'environnement n'est pas définie.

Amazon S3

Cette variable n'est pas définie.

Le cas échéant, la `CODEBUILD_RESOLVED_SOURCE_VERSION` variable n'est disponible qu'après la `DOWNLOAD_SOURCE` phase.

`CODEBUILD_SOURCE_REPO_URL`

URL de l'artefact ou du référentiel de code source d'entrée. Pour Amazon S3, il est `s3://` suivi du nom du compartiment et du chemin d'accès à l'artefact d'entrée. Pour CodeCommit et GitHub, il s'agit de l'URL du clone du dépôt. Si un build provient de CodePipeline, cette variable d'environnement peut être vide.

Pour les sources secondaires, la variable d'environnement pour l'URL du référentiel source secondaire est `CODEBUILD_SOURCE_REPO_URL_<sourceIdentifiant>`, où *<sourceIdentifiant>* est l'identifiant de source que vous créez.

`CODEBUILD_SOURCE_VERSION`

Le format de la valeur dépend du référentiel source.

- Pour Amazon S3, il s'agit de l'ID de version associé à l'artefact d'entrée.
- En CodeCommit effet, il s'agit de l'ID de validation ou du nom de branche associé à la version du code source à compiler.
- Pour GitHub GitHub Enterprise Server et Bitbucket, il s'agit de l'ID de validation, du nom de branche ou du nom de balise associés à la version du code source à créer.

Note

Pour une version GitHub ou une version de serveur GitHub d'entreprise déclenchée par un événement de pull request webhook, c'est par */pull-request-number* le cas.

Pour les sources secondaires, la variable d'environnement de la version de la source secondaire est `CODEBUILD_SOURCE_VERSION_<sourceIdentifiant>`, où *<sourceIdentifiant>* est l'identifiant de source que vous créez. Pour de plus amples informations, veuillez consulter [Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#).

`CODEBUILD_SRC_DIR`

Le chemin du répertoire CodeBuild utilisé pour la construction (par exemple, `/tmp/src123456789/src`).

Pour les sources secondaires, la variable d'environnement pour le chemin du répertoire source secondaire est `CODEBUILD_SRC_DIR_<sourceIdentifiant>`, où se *<sourceIdentifiant>* trouve l'identifiant de source que vous créez. Pour de plus amples informations, veuillez consulter [Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#).

CODEBUILD_START_TIME

Heure de début de la construction spécifiée sous forme d'horodatage Unix en millisecondes.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

L'ID de compte de l'utilisateur qui a déclenché l'événement webhook.

CODEBUILD_WEBHOOK_BASE_REF

Le nom de référence de base de l'événement webhook qui déclenche la version en cours. Pour une demande d'extraction, c'est ce que l'on appelle la référence de branche.

CODEBUILD_WEBHOOK_EVENT

L'événement webhook qui déclenche la version en cours.

CODEBUILD_WEBHOOK_MERGE_COMMIT

Identifiant du commit de fusion utilisé pour le build. Cette variable est définie lorsqu'une pull request Bitbucket est fusionnée avec la stratégie squash et que la branche pull request est fermée. Dans ce cas, le commit de la pull request d'origine n'existe plus. Cette variable d'environnement contient donc l'identifiant du commit de fusion écrasé.

CODEBUILD_WEBHOOK_PREV_COMMIT

L'ID du dernier commit avant l'événement push du webhook qui déclenche la version en cours.

CODEBUILD_WEBHOOK_HEAD_REF

Le nom de référence principal de l'événement webhook qui déclenche la version en cours. Il peut s'agir d'une référence de branche ou d'une référence de balise.

CODEBUILD_WEBHOOK_TRIGGER

Affiche l'événement webhook qui a déclenché le build. Cette variable est disponible uniquement pour les builds déclenchés par un webhook. La valeur est analysée à partir de la charge utile envoyée CodeBuild à GitHub GitHub Enterprise Server ou Bitbucket. Le format de la valeur dépend de quel type d'événement a déclenché le build.

- Pour les builds déclenchés par une demande d'extraction, c'est `pr/pull-request-number`.

- Pour les builds déclenchés par la création d'une nouvelle branche ou par l'envoi d'un engagement à une branche, c'est `branch/branch-name`.
- Pour les builds déclenchés par la transmission d'une balise à un référentiel, c'est `tag/tag-name`.

MAISON

Cette variable d'environnement est toujours définie sur `/root`.

AWS CodeBuild prend également en charge un ensemble de variables d'environnement pour les versions de coureurs auto-hébergées. Pour en savoir plus sur les coureurs CodeBuild auto-hébergés, consultez [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

CODEBUILD_RUNNER_OWNER

Le propriétaire du référentiel qui déclenche la compilation auto-hébergée par Runner.

CODEBUILD_RUNNER_REPO

Le nom du dépôt qui déclenche la compilation auto-hébergée par Runner.

CODEBUILD_RUNNER_REPO_DOMAIN

Le domaine du référentiel qui déclenche la compilation auto-hébergée par le runner. Uniquement les versions GitHub d'entreprise spécifiées.

CODEBUILD_WEBHOOK_LABEL

L'étiquette utilisée pour configurer les remplacements de build et le runner auto-hébergé pendant la construction.

CODEBUILD_WEBHOOK_RUN_ID

ID d'exécution du flux de travail associé à la compilation.

CODEBUILD_WEBHOOK_JOB_ID

L'ID de tâche de la tâche associée à la version.

CODEBUILD_WEBHOOK_WORKFLOW_NAME

Le nom du flux de travail associé au build s'il existe dans la charge utile de la demande de webhook.

CODEBUILD_RUNNER_WITH_BUILDSPEC

Si une dérogation buildspec est configurée dans les étiquettes de demande d'exécution auto-hébergées, elle est définie sur `true`

Vous pouvez également fournir des environnements de génération avec vos propres variables d'environnement. Pour plus d'informations, consultez les rubriques suivantes :

- [À utiliser CodeBuild avec CodePipeline](#)
- [Création d'un projet de génération](#)
- [Modifier les paramètres du projet de construction](#)
- [Exécuter les builds manuellement](#)
- [Référence des spécifications de génération](#)

Pour répertorier toutes les variables d'environnement disponibles dans un environnement de génération, vous pouvez exécuter la commande `printenv` (pour les environnements de génération basés sur Linux) ou `"Get-ChildItem Env:"` (les environnements de génération basés sur Windows) pendant une génération. À l'exception de celles répertoriées précédemment, les variables d'environnement commençant par `CODEBUILD_` sont destinées à un usage CodeBuild interne. Elles ne doivent pas être utilisées dans vos commandes de génération.

Important

Nous déconseillons fortement l'utilisation de variables d'environnement pour stocker des valeurs sensibles, en particulier les clés AWS IDs d'accès. Les variables d'environnement peuvent être affichées en texte brut à l'aide d'outils tels que la CodeBuild console et le AWS CLI.

Nous vous recommandons de stocker les valeurs sensibles dans le magasin de paramètres Amazon EC2 Systems Manager, puis de les récupérer à partir de vos spécifications de construction. Pour stocker des valeurs sensibles, consultez la section [Stockage des paramètres de Systems Manager](#) et [procédure pas à pas : création et test d'un paramètre de chaîne \(console\)](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager. Pour les récupérer, consultez le mappage `parameter-store` dans [Syntaxe d'un fichier buildspec](#).

CODEBUILD_BUILD_URL

URL des résultats de compilation pour cette version.

Tâches en arrière-plan dans les environnements de génération

Vous pouvez exécuter des tâches en arrière-plan dans les environnements de génération. Pour ce faire, dans vos spécifications de génération, utilisez la commande `nohup` pour exécuter une commande en tant que tâche en arrière-plan, même si le processus de génération quitte le shell. Utilisez la commande `disown` pour forcer l'arrêt d'une tâche en arrière-plan en cours d'exécution.

Exemples :

- Démarrer un processus en arrière-plan et attendre que celui-ci se termine ultérieurement :

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Démarrer un processus en arrière-plan et ne pas attendre que celui-ci se termine :

```
|  
nohup sleep 30 & disown $!
```

- Démarrer un processus en arrière-plan et y mettre fin ultérieurement :

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
kill $(cat pidfile)
```

Construisez des projets

Un projet de build inclut des informations sur la façon d'exécuter une compilation, notamment sur l'emplacement du code source, sur l'environnement de compilation à utiliser, sur les commandes de compilation à exécuter et sur l'emplacement où stocker le résultat de la compilation.

Vous pouvez effectuer ces tâches lorsque vous gérez des projets de génération :

Rubriques

- [Créez un projet de construction dans AWS CodeBuild](#)
- [Création d'une règle de notification](#)
- [Modifier les paramètres du projet de construction dans AWS CodeBuild](#)
- [Plusieurs jetons d'accès dans CodeBuild](#)
- [Supprimer des projets de construction dans AWS CodeBuild](#)
- [Obtenir un projet de construction public URLs](#)
- [Partagez des projets de construction](#)
- [Tag : projets de construction](#)
- [Utilisez des coureurs avec AWS CodeBuild](#)
- [Utilisez des webhooks avec AWS CodeBuild](#)
- [Afficher les détails d'un projet de construction dans AWS CodeBuild](#)
- [Afficher les noms des projets de construction dans AWS CodeBuild](#)

Créez un projet de construction dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs créer un projet de construction. AWS CLI

Rubriques

- [Prérequis](#)
- [Création d'un projet de génération \(console\)](#)
- [Création d'un projet de génération \(AWS CLI\)](#)
- [Création d'un projet de génération \(AWS SDKs\)](#)
- [Création d'un projet de génération \(CloudFormation\)](#)

Prérequis

Avant de créer un projet de construction, répondez aux questions posées dans [Planification d'une génération](#).

Création d'un projet de génération (console)

Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.

Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

Choisissez Créer un projet de génération.

Remplissez les sections suivantes. Une fois terminé, choisissez Créer un projet de construction au bas de la page.

Rubriques :

- [Configuration du projet](#)
- [Source](#)
- [Environnement](#)
- [Spécifications de construction](#)
- [Configuration par lots](#)
- [Artefacts](#)
- [Journaux](#)

Configuration du projet

Nom du projet

Entrez un nom pour ce projet de construction. Les noms des projets de construction doivent être uniques pour chaque AWS compte.

Description

Entrez une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

Construire un badge

(Facultatif) Sélectionnez Activer le badge de construction pour rendre l'état de construction de votre projet visible et intégrable. Pour de plus amples informations, veuillez consulter [Exemple de badges de génération](#).

Note

Le badge de construction ne s'applique pas si votre fournisseur source est Amazon S3.

Activer la limite de génération simultanée

(Facultatif) Si vous souhaitez limiter le nombre de builds simultanés pour ce projet, effectuez les opérations suivantes :

1. Sélectionnez Restreindre le nombre de versions simultanées que ce projet peut démarrer.
2. Dans Limite de génération simultanée, entrez le nombre maximum de versions simultanées autorisées pour ce projet. Cette limite ne peut pas être supérieure à la limite de création simultanée définie pour le compte. Si vous essayez de saisir un nombre supérieur à la limite du compte, un message d'erreur s'affiche.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Informations supplémentaires

(Facultatif) Pour Tags, entrez le nom et la valeur de tous les tags que vous souhaitez que les AWS services d'assistance utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.

Source

Fournisseur de source

Choisissez le type de fournisseur de code source. Utilisez les listes suivantes pour effectuer des sélections adaptées à votre fournisseur de source :

 Note

CodeBuild ne prend pas en charge Bitbucket Server.

Amazon S3

Compartiment

Choisissez le nom du compartiment d'entrée contenant le code source.

Clé d'objet S3 ou dossier S3

Entrez le nom du fichier ZIP ou le chemin d'accès au dossier contenant le code source. Entrez une barre oblique (/) pour tout télécharger dans le compartiment S3.

Version de la source

Entrez l'ID de version de l'objet qui représente la version de votre fichier d'entrée. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

CodeCommit

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Type de référence

Choisissez Branch, Git tag ou Commit ID pour spécifier la version de votre code source. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez de créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Bitbucket

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez CodeConnectionsle OAuthmot de passe de l'application ou le jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild.

Connexion

Sélectionnez une connexion Bitbucket ou un secret de Secrets Manager pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository dans mon compte Bitbucket ou Public repository et saisissez l'URL du référentiel.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le name paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

Pour l'URL cible, entrez la valeur à utiliser pour le url paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [Événements du webhook Bitbucket](#).

GitHub

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez GitHub Application ou jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild. OAuth

Connexion

Sélectionnez une GitHub connexion ou un secret du Gestionnaire de Secrets pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository in my GitHub account, Public repository ou GitHub Scoped Webhook et entrez l'URL du référentiel.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez **Use Git submodules** (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez **Signaler les statuts de construction** au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte d'état, entrez la valeur à utiliser pour le `context` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour l'URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans **Événements de webhook source primaire**, sélectionnez **Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel** si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitHub Enterprise Server

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez CodeConnections un jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild.

Connexion

Sélectionnez une connexion GitHub Enterprise ou un secret Secrets Manager pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository in my GitHub Enterprise account ou GitHub Enterprise Scoped Webhook et entrez l'URL du référentiel.

Version de la source

Entrez une pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez **Use Git submodules** (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez **Signaler les statuts de construction** au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte d'état, entrez la valeur à utiliser pour le `context` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour l'URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

SSL non sécurisé

Sélectionnez **Activer le protocole SSL non sécurisé** pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub d'entreprise.

Dans **Événements de webhook source primaire**, sélectionnez **Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel** si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitLab

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

CodeConnection est utilisé pour se connecter GitLab à CodeBuild.

Connexion

Sélectionnez une GitLab connexion par laquelle vous souhaitez vous connecter CodeConnections.

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Version de la source

Entrez un identifiant de pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

GitLab Self Managed

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

CodeConnection est utilisé pour connecter GitLab Self Managed à CodeBuild.

Connexion

Sélectionnez une connexion GitLab autogérée par laquelle vous souhaitez vous connecter CodeConnections.

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Version de la source

Entrez un identifiant de pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d

ou5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Environnement

Modèle de provisionnement

Effectuez l'une des actions suivantes :

- Pour utiliser des flottes à la demande gérées par AWS CodeBuild, choisissez On-Demand. Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites une fois la construction terminée. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.
- Pour utiliser des flottes de capacité réservée gérées par AWS CodeBuild, choisissez Capacité réservée, puis sélectionnez un nom de flotte. Avec les flottes de capacité réservée, vous configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

Pour plus d'informations, veuillez consulter [Exécutez des builds sur des flottes à capacité réservée](#).

Image de l'environnement

Effectuez l'une des actions suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Calcul

Effectuez l'une des actions suivantes :

- Pour utiliser le EC2 calcul, choisissez EC2. EC2 le calcul offre une flexibilité optimisée lors des exécutions d'actions.
- Pour utiliser le calcul Lambda, choisissez Lambda. Le calcul Lambda offre des vitesses de démarrage optimisées pour vos builds. Lambda permet des builds plus rapides grâce à une latence de démarrage plus faible. Lambda s'adapte également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Pour plus d'informations, veuillez consulter [Exécuter des builds sur AWS Lambda ordinateur](#).

Rôle de service

Effectuez l'une des actions suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

Note

Lorsque vous utilisez la console pour créer un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

Configuration supplémentaire

Limite de tentatives automatiques

Spécifiez le nombre de tentatives automatiques supplémentaires après un échec de compilation. Par exemple, si la limite de tentatives automatiques est définie sur 2, l'API CodeBuild appellera l'`RetryBuildAPI` pour réessayer automatiquement votre build jusqu'à 2 fois supplémentaires.

Expiration

Spécifiez une valeur, comprise entre 5 minutes et 36 heures, après quoi la CodeBuild génération s'arrête si elle n'est pas terminée. Si les valeurs de heures et minutes sont laissées vides, la valeur par défaut de 60 minutes est utilisée.

privilegié

(Facultatif) Sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos versions obtiennent des privilèges élevés uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre spécification de génération en exécutant les commandes de génération ci-après. N'exécutez

pas ces commandes si vous avez choisi une image d'environnement de construction fournie CodeBuild par le support Docker.

 Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux](#) Capabilities sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources du VPCs

Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Calcul

Choisissez l'une des options disponibles.

Informations d'identification du registre

Spécifiez un identifiant de registre lorsque le projet est configuré avec une image de registre non privée.

 Note

Ces informations d'identification ne seront utilisées que si les images sont remplacées par celles provenant de registres privés.

Variables d'environnement

Entrez le nom et la valeur, puis choisissez le type de chaque variable d'environnement à utiliser pour les builds.

Note

CodeBuild définit automatiquement la variable d'environnement pour votre AWS région. Vous devez définir les variables d'environnement suivantes si vous ne les avez pas ajoutées dans votre fichier `buildspec.yml` :

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

La console et AWS CLI les utilisateurs peuvent voir les variables d'environnement. Si la visibilité de vos variables d'environnement ne vous pose pas de problème, définissez les zones Nom et Valeur, puis définissez Type sur Texte brut.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un identifiant de clé d' AWS accès, une clé d'accès AWS secrète ou un mot de passe en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store ou AWS Secrets Manager.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, choisissez Parameter dans Type. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value, entrez le nom du paramètre tel qu'il est enregistré dans Amazon EC2 Systems Manager Parameter Store. Si l'on prend comme exemple un paramètre nommé `/CodeBuild/dockerLoginPassword`, pour Type, choisissez Parameter (Paramètre). Pour Nom, saisissez `LOGIN_PASSWORD`. Pour le champ Valeur, saisissez `/CodeBuild/dockerLoginPassword`.

Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez Create parameter (Créer un paramètre), puis suivez

les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre /CodeBuild/ tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser l'`ssm:GetParameters` action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de construction fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par /CodeBuild/, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de paramètres qui ne commencent pas par /CodeBuild/. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par /CodeBuild/.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres de l'espace de /CodeBuild/ noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée MY_VAR avec la valeur my_value et que vous définissez une variable d'environnement nommée MY_VAR avec la valeur other_value, la valeur my_value est remplacée par other_value. De même, si l'image Docker contient déjà une variable d'environnement nommée PATH avec la valeur /usr/local/sbin:/usr/local/bin et que vous définissez une variable d'environnement nommée PATH avec la valeur \$PATH:/usr/share/ant/bin, la valeur /usr/local/sbin:/usr/local/bin est remplacée par la valeur littérale \$PATH:/usr/share/ant/bin.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration `buildspec` a la priorité la plus faible.

Si vous utilisez Secrets Manager, pour Type, choisissez Secrets Manager. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value (Valeur), saisissez un `reference-key` à l'aide du modèle `secret-id:json-key:version-stage:version-id`. Pour plus d'informations, veuillez consulter [Secrets Manager reference-key in the buildspec file](#).

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser `secretsmanager:GetSecretValue` action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par `/CodeBuild/`. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de /CodeBuild/ noms dans le Gestionnaire de secrets.

Spécifications de construction

Spécifications de construction

Effectuez l'une des actions suivantes :

- Si votre code source inclut un fichier buildspec, choisissez Utiliser un fichier buildspec. Par défaut, CodeBuild recherche un fichier nommé `buildspec.yml` dans le répertoire racine de code source. Si votre fichier buildspec utilise un nom ou un emplacement différent, entrez son chemin depuis la racine source dans le nom Buildspec (par exemple, `ou. buildspec-two.yml configuration/buildspec.yml`). Si le fichier buildspec se trouve dans un compartiment S3, il doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier buildspec à l'aide de son ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Si votre code source ne comprend pas de fichier de spécification de génération ou si vous souhaitez exécuter des commandes de génération différentes de celles spécifiées pour la phase build dans le fichier `buildspec.yml` au sein du répertoire racine du code source, choisissez Insérer des commandes de génération. Pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase build. Pour plusieurs commandes, séparez celles-ci avec `&&` (par exemple, `mvn test && mvn package`). Pour exécuter des commandes dans d'autres phases, ou si vous avez une longue liste de commandes pour la build phase, ajoutez un `buildspec.yml` fichier dans le répertoire racine du code source, ajoutez les commandes au fichier, puis choisissez Utiliser le fichier `buildspec.yml` dans le répertoire racine du code source.

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

Configuration par lots

Vous pouvez exécuter un groupe de builds en une seule opération. Pour de plus amples informations, veuillez consulter [Exécuter des builds par lots](#).

Définition de la configuration par lots

Sélectionnez cette option pour autoriser les compilations par lots dans ce projet.

Rôle du service Batch

Fournit le rôle de service pour les compilations par lots.

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle de service par lots, choisissez Nouveau rôle de service. Dans Rôle de service, entrez le nom du nouveau rôle.
- Si vous avez un rôle de service par lots, choisissez Rôle de service existant. Dans Rôle de service, choisissez le rôle de service.

Les builds par lots introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est requis car CodeBuild il doit être capable d'appeler les RetryBuild actions StartBuildStopBuild, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même que celui qu'ils utilisent dans leur build, pour deux raisons :

- L'attribution du rôle StartBuild de construction et RetryBuild des autorisations permettrait à une seule version de démarrer d'autres versions via le buildspec. StopBuild
- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds eux-mêmes puissent contourner ces restrictions.

Types de calcul autorisés pour le traitement par lots

Sélectionnez les types de calcul autorisés pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Flottes autorisées pour le lot

Sélectionnez les flottes autorisées pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Nombre maximal de builds autorisés par lot

Entrez le nombre maximum de builds autorisés dans le lot. Si un lot dépasse cette limite, il échouera.

Délai d'expiration du Batch

Entrez la durée maximale pendant laquelle la génération par lots doit être terminée.

Combinez des artefacts

Sélectionnez Combiner tous les artefacts du lot en un seul emplacement pour que tous les artefacts du lot soient combinés en un seul emplacement.

Mode de rapport par lots

Sélectionnez le mode de rapport d'état de construction souhaité pour les versions par lots.

Note

Ce champ n'est disponible que lorsque la source du projet est Bitbucket ou GitHub Enterprise GitHub, et l'option Signaler les statuts de construction au fournisseur de source lorsque le début et la fin de vos builds sont sélectionnés sous Source.

Constructions agrégées

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient combinés dans un seul rapport d'état.

Constructions individuelles

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient signalés séparément.

Artefacts

Type

Effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer des artefacts de sortie de génération, choisissez Aucun artefact. Vous pouvez le faire si vous exécutez uniquement des tests de compilation ou si vous souhaitez transférer une image Docker vers un référentiel Amazon ECR.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. (Si vous souhaitez produire un fichier ZIP et que vous voulez que celui-ci ait une extension de fichier, veillez à l'inclure après le nom de fichier ZIP.)

- Sélectionnez Activer la gestion sémantique des versions si vous voulez qu'un nom spécifié dans le fichier buildspec remplace le nom spécifié dans la console. Le nom figurant dans un fichier buildspec est calculé au moment de la génération et utilise le langage de commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).
- Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
- Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).
- Si vous ne souhaitez pas que vos artefacts de génération soient chiffrés, choisissez Remove artifacts encryption (Supprimer le chiffrement des artefacts).

Pour chaque ensemble d'artefacts secondaire que vous souhaitez :

1. Pour Artifact identifier (Identifiant d'artefact), saisissez une valeur de moins de 128 caractères et contenant uniquement des caractères alphanumériques et des traits de soulignement.
2. Choisissez Add artifact (Ajouter un artefact).
3. Suivez les étapes précédentes pour configurer vos artefacts secondaires.
4. Choisissez Save artifact (Enregistrer l'artefact).

Configuration supplémentaire

Clé de chiffrement

(Facultatif) Effectuez l'une des actions suivantes :

- Pour utiliser le Clé gérée par AWS for Amazon S3 dans votre compte afin de chiffrer les artefacts de sortie du build, laissez la clé de chiffrement vide. Il s'agit de l'option par défaut.
- Pour utiliser une clé gérée par le client pour chiffrer les artefacts de sortie de génération, dans Clé de chiffrement, entrez l'ARN de la clé KMS. Utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Type de cache

Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :

- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

 Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

 Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Des mises en cache pour améliorer les performances](#).

Journaux

Choisissez les journaux que vous souhaitez créer. Vous pouvez créer des CloudWatch journaux Amazon Logs, des journaux Amazon S3 ou les deux.

CloudWatch

Si vous souhaitez obtenir CloudWatch les journaux Amazon Logs :

CloudWatch journaux

Sélectionnez Journaux CloudWatch .

Nom du groupe

Entrez le nom de votre groupe de CloudWatch journaux Amazon Logs.

Nom du stream

Entrez le nom de votre flux de journal Amazon CloudWatch Logs.

S3

Si vous souhaitez obtenir les journaux Amazon S3 :

Journaux S3

Sélectionnez Journaux S3.

Compartiment

Choisissez le nom du compartiment S3 pour vos journaux.

Préfixe de chemin

Entrez le préfixe de vos journaux.

Désactiver le chiffrement des journaux S3

Sélectionnez si vous ne voulez pas que vos journaux S3 soient chiffrés.

Création d'un projet de génération (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with CodeBuild, consultez le [Référence des commandes en ligne](#).

Pour créer un projet de CodeBuild construction à l'aide de AWS CLI, vous devez créer une structure de [projet](#) au format JSON, renseigner la structure et appeler la [create-project](#) commande pour créer le projet.

Créez le fichier JSON

Créez un fichier JSON squelette à l'aide de la [create-project](#) commande, en utilisant l'option `--generate-cli-skeleton` :

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Cela crée un fichier JSON avec le chemin et le nom de fichier spécifiés par *<json-file>*.

Renseignez le fichier JSON

Modifiez les données JSON comme suit et enregistrez vos résultats.

```
{
  "name": "<project-name>",
  "description": "<description>",
  "source": {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" | "GITLAB" |
    "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "buildStatusConfig": {
      "context": "<context>",
      "targetUrl": "<target-url>"
    },
    "gitSubmodulesConfig": {
      "fetchSubmodules": "<fetch-submodules>"
    },
    "auth": {
      "type": "<auth-type>",
      "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
      "GITLAB" | "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
      "location": "<source-location>",
      "gitCloneDepth": "<git-clone-depth>",
      "buildspec": "<buildspec>",
      "InsecureSsl": "<insecure-ssl>",
      "reportBuildStatus": "<report-build-status>",
      "auth": {
        "type": "<auth-type>",
        "resource": "<auth-resource>"
      },
      "sourceIdentifier": "<source-identifier>"
    }
  ]
}
```



```

    }
  ],
  "secondarySourceVersions": [
    {
      "sourceIdentifier": "<secondary-source-identifier>",
      "sourceVersion": "<secondary-source-version>"
    }
  ],
  "sourceVersion": "<source-version>",
  "artifacts": {
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<artifacts-location>",
    "path": "<artifacts-path>",
    "namespaceType": "<artifacts-namespacetype>",
    "name": "<artifacts-name>",
    "overrideArtifactName": "<override-artifact-name>",
    "packaging": "<artifacts-packaging>"
  },
  "secondaryArtifacts": [
    {
      "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
      "location": "<secondary-artifact-location>",
      "path": "<secondary-artifact-path>",
      "namespaceType": "<secondary-artifact-namespaceType>",
      "name": "<secondary-artifact-name>",
      "packaging": "<secondary-artifact-packaging>",
      "artifactIdentifier": "<secondary-artifact-identifier>"
    }
  ],
  "cache": {
    "type": "<cache-type>",
    "location": "<cache-location>",
    "mode": [
      "<cache-mode>"
    ]
  },
  "environment": {
    "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
    "WINDOWS_SERVER_2019_CONTAINER" | "WINDOWS_SERVER_2022_CONTAINER",
    "image": "<image>",
    "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
    "BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
    "certificate": "<certificate>",
    "environmentVariables": [

```

```

    {
      "name": "<environmentVariable-name>",
      "value": "<environmentVariable-value>",
      "type": "<environmentVariable-type>"
    }
  ],
  "registryCredential": [
    {
      "credential": "<credential-arn-or-name>",
      "credentialProvider": "<credential-provider>"
    }
  ],
  "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
  "privilegedMode": "<privileged-mode>"
},
"serviceRole": "<service-role>",
"autoRetryLimit": <auto-retry-limit>,
"timeoutInMinutes": <timeout>,
"queuedTimeoutInMinutes": <queued-timeout>,
"encryptionKey": "<encryption-key>",
"tags": [
  {
    "key": "<tag-key>",
    "value": "<tag-value>"
  }
],
"vpcConfig": {
  "securityGroupIds": [
    "<security-group-id>"
  ],
  "subnets": [
    "<subnet-id>"
  ],
  "vpcId": "<vpc-id>"
},
"badgeEnabled": "<badge-enabled>",
"logsConfig": {
  "cloudWatchLogs": {
    "status": "<cloudwatch-logs-status>",
    "groupName": "<group-name>",
    "streamName": "<stream-name>"
  },
  "s3Logs": {
    "status": "<s3-logs-status>",

```

```
    "location": "<s3-logs-location>",
    "encryptionDisabled": "<s3-logs-encryption-disabled>"
  },
  "fileSystemLocations": [
    {
      "type": "EFS",
      "location": "<EFS-DNS-name-1>:/<directory-path>",
      "mountPoint": "<mount-point>",
      "identifier": "<efs-identifier>",
      "mountOptions": "<efs-mount-options>"
    }
  ],
  "buildBatchConfig": {
    "serviceRole": "<batch-service-role>",
    "combineArtifacts": <combine-artifacts>,
    "restrictions": {
      "maximumBuildsAllowed": <max-builds>,
      "computeTypesAllowed": [
        "<compute-type>"
      ],
      "fleetsAllowed": [
        "<fleet-name>"
      ]
    },
    "timeoutInMins": <batch-timeout>,
    "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
  },
  "concurrentBuildLimit": <concurrent-build-limit>
}
```

Remplacez les éléments suivants :

nom

Obligatoire. Nom de ce projet de génération. Ce nom doit être unique pour tous les projets de construction de votre AWS compte.

description

Facultatif. Description de ce projet de génération.

source

Obligatoire. Un [ProjectSource](#) objet qui contient des informations sur les paramètres du code source de ce projet de build. Après avoir ajouté un objet source, vous pouvez ajouter jusqu'à 12 autres sources à l'aide de . Il s'agit notamment des paramètres suivants :

source/type

Obligatoire. Type de référentiel qui contient le code source à générer. Les valeurs valides sont les suivantes :

- CODECOMMIT
- CODEPIPELINE
- GITHUB
- GITHUB_ENTERPRISE
- GITLAB
- GITLAB_SELF_MANAGED
- BITBUCKET
- S3
- NO_SOURCE

Si vous utilisez NO_SOURCE, buildspec ne peut pas être un fichier, car le projet n'a pas de source. En revanche, vous devez utiliser l'attribut buildspec afin de spécifier une chaîne au format YAML pour votre buildspec. Pour de plus amples informations, veuillez consulter [Création d'un projet de construction sans source](#).

source/ lieu

Obligatoire sauf si vous avez défini *<source-type>* sur CODEPIPELINE. Emplacement du code source pour le type de référentiel spécifié.

- Pour CodeCommit, l'URL du clone HTTPS vers le référentiel qui contient le code source et le fichier buildspec (par exemple,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Pour Amazon S3, le nom du compartiment d'entrée de construction, suivi du chemin et du nom du fichier ZIP contenant le code source et les spécifications de construction. Par exemple :
 - Pour un fichier ZIP situé à la racine du compartiment d'entrée : *<bucket-name>/<object-name>.zip*.

- Pour un fichier ZIP situé dans un sous-dossier du compartiment d'entrée : `<bucket-name>/<subfolder-path>/<object-name>.zip`.
- Pour GitHub, l'URL du clone HTTPS vers le référentiel contenant le code source et le fichier buildspec. L'URL doit contenir github.com. Vous devez connecter votre AWS compte à votre GitHub compte. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.
 - Choisissez Authorize application. (Une fois connecté à votre GitHub compte, vous n'avez pas besoin de terminer la création du projet de construction. Vous pouvez fermer la CodeBuild console.)
- Pour GitHub Enterprise Server, URL du clone HTTP ou HTTPS du référentiel contenant le code source et le fichier buildspec. Vous devez également connecter votre AWS compte à votre compte GitHub Enterprise Server. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.
 1. Créez un jeton d'accès personnel dans GitHub Enterprise Server.
 2. Copiez ce jeton dans votre presse-papiers afin de pouvoir l'utiliser lors de la création de votre CodeBuild projet. Pour plus d'informations, consultez la section [Création d'un jeton d'accès personnel pour la ligne de commande](#) sur le site Web d' GitHub aide.
 3. Lorsque vous utilisez la console pour créer votre CodeBuild projet, dans Source, pour Source provider, choisissez GitHubEnterprise.
 4. Pour Personal Access Token, collez le jeton copié dans le Presse-papiers. Choisissez Save Token. Votre CodeBuild compte est désormais connecté à votre compte GitHub Enterprise Server.
- Pour GitLab et GitLab autogéré, l'URL du clone HTTPS vers le référentiel contenant le code source et le fichier buildspec. Notez que si vous l'utilisez GitLab, l'URL doit contenir gitlab.com. Si vous utilisez l' GitLab autogéré, l'URL ne doit pas nécessairement contenir gitlab.com. Vous devez connecter votre AWS compte à votre compte GitLab ou à votre compte GitLab autogéré. Pour cela, utilisez la console CodeBuild pour créer un projet de génération.
 - Dans le volet de navigation des outils de développement, choisissez Paramètres, Connexions, puis Créer une connexion. Sur cette page, créez une connexion GitLab ou une connexion GitLab autogérée, puis choisissez Connect to GitLab.
- Pour Bitbucket, l'URL de clone HTTPS vers le référentiel qui contient le code source et le fichier buildspec. L'URL doit contenir bitbucket.org. Vous devez également connecter votre AWS compte à votre compte Bitbucket. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.

1. Lorsque vous utilisez la console pour vous connecter (ou vous reconnecter) à Bitbucket, dans la page Bitbucket Confirm access to your account, choisissez Grant access. (Une fois connecté à votre compte Bitbucket, vous n'avez pas besoin de terminer la création du projet de build. Vous pouvez fermer la CodeBuild console.)
- Pour AWS CodePipeline, ne spécifiez pas de location valeur pour source. CodePipeline ignore cette valeur car lorsque vous créez un pipeline dans CodePipeline, vous spécifiez l'emplacement du code source dans le stage Source du pipeline.

source/ gitCloneDepth

Facultatif. Profondeur de l'historique à télécharger. La valeur minimum est de 0. Si cette valeur est égale à 0, supérieure à 25 ou non spécifiée, l'historique complet est téléchargé avec chaque projet de génération. Si votre type de source est Amazon S3, cette valeur n'est pas prise en charge.

source/ buildspec

Facultatif. Définition de spécification de génération ou fichier à utiliser. Si cette valeur n'est pas fournie ou qu'elle est définie sur une chaîne vide, le code source doit contenir un fichier `buildspec.yml` dans son répertoire racine. Si cette valeur est définie, il peut s'agir d'une définition de buildspec intégrée, du chemin d'accès à un autre fichier buildspec relatif au répertoire racine de votre source principale ou du chemin d'accès à un compartiment S3. Le bucket doit se trouver dans la même AWS région que le projet de construction. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Pour de plus amples informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).

source/ authentication

Contient des informations sur les paramètres d'autorisation CodeBuild pour accéder au code source à créer.

source/ auteur/type

Obligatoire. Le type d'autorisation à utiliser. Les valeurs valides sont :

- OAUTH
- CODECONNECTIONS
- SECRETS_MANAGER

source/auteur/ressource

Facultatif. La valeur de la ressource qui s'applique au type d'autorisation spécifié. Il peut s'agir de l'ARN ou de l'ARN CodeConnections du Secrets Manager.

source/ reportBuildStatus

Indique si vous souhaitez envoyer le statut de début et de fin d'une génération à votre fournisseur de source. Si vous le définissez avec un fournisseur de source autre qu'GitHub GitHub Enterprise Server ou Bitbucket, un `invalidInputException` est émis.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

source/ buildStatusConfig

Contient des informations qui définissent la manière dont le CodeBuild projet de construction communique l'état de la construction au fournisseur source. Cette option n'est utilisée que lorsque le type de source est `GITHUBGITHUB_ENTERPRISE`, ou `BITBUCKET`.

source/buildStatusConfig/contexte

Pour les sources Bitbucket, ce paramètre est utilisé pour le `name` paramètre figurant dans le statut de validation de Bitbucket. Pour les GitHub sources, ce paramètre est utilisé pour le `context` paramètre figurant dans l'état de GitHub validation.

Par exemple, vous pouvez faire en sorte qu'ils `context` contiennent le numéro de version et le déclencheur webhook à l'aide des variables d' environnement CodeBuild :

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -  
${CODEBUILD_WEBHOOK_TRIGGER}
```

Il en résulte que le contexte apparaît comme suit pour la version #24 déclenchée par un événement de pull request du webhook :

```
AWS CodeBuild sample-project Build #24 - pr/8
```

source/buildStatusConfig/URL cible

Pour les sources Bitbucket, ce paramètre est utilisé pour le `url` paramètre figurant dans le statut de validation de Bitbucket. Pour les GitHub sources, ce paramètre est utilisé pour le `target_url` paramètre figurant dans l'état de GitHub validation.

Par exemple, vous pouvez `targetUrl` définir le statut sur `https://aws.amazon.com/codebuild/<path to build>` et le statut de validation sera lié à cette URL.

Vous pouvez également inclure des variables d' CodeBuild environnement dans le `targetUrl` pour ajouter des informations supplémentaires à l'URL. Par exemple, pour ajouter la région de construction à l'URL, définissez la valeur `targetUrl` sur :

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Si la région de construction l'est `us-east-2`, elle s'étendra à :

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

source/ gitSubmodulesConfig

Facultatif. Informations sur la configuration des sous-modules Git. Utilisé uniquement avec CodeCommit GitHub, GitHub Enterprise Server et Bitbucket.

source/gitSubmodulesConfig/FetchSubmodules

Définissez `fetchSubmodules` sur `true` si vous souhaitez inclure les sous-modules Git dans votre référentiel. Les sous-modules Git inclus doivent être configurés en tant que HTTPS.

source/ InsecureSsl

Facultatif. Utilisé uniquement avec GitHub Enterprise Server. Définissez cette valeur sur `true` pour ignorer les avertissements TLS lors de la connexion au référentiel de votre projet GitHub Enterprise Server. La valeur par défaut est `false`. `InsecureSsl` doit être utilisé à des fins de test uniquement. Cette option ne doit pas être utilisée dans un environnement de production.

source/ Identifiant de source

Identifiant défini par l'utilisateur pour la source du projet. Facultatif pour la source principale. Nécessaire pour les sources secondaires.

Sources secondaires

Facultatif. Tableau d'[ProjectSource](#) objets contenant des informations sur les sources secondaires d'un projet de construction. Vous pouvez ajouter jusqu'à 12 sources secondaires. Les `secondarySources` objets utilisent les mêmes propriétés que celles utilisées par l'objet. Dans un objet source secondaire, le `sourceIdentifier` est obligatoire.

`secondarySourceVersions`

Facultatif. Tableau d'objets [ProjectSourceVersion](#). Si `secondarySourceVersions` est spécifié au niveau de la build, ils ont priorité sur cela.

Version de la source

Facultatif. Version de l'entrée de compilation à créer pour ce projet. Si elle n'est pas spécifiée, la dernière version est utilisée. Si cette valeur est spécifiée, elle doit être l'une des suivantes :

- Pour CodeCommit, l'ID de validation, la branche ou le tag Git à utiliser.
- Pour GitHub, l'ID de validation, l'ID de pull request, le nom de branche ou le nom de balise correspondant à la version du code source que vous souhaitez créer. Si un ID de demande d'extraction est spécifié, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Pour GitLab, l'ID de validation, l'ID de pull request, le nom de la branche, le nom du tag ou la référence et un ID de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).
- Pour Bitbucket, l'ID de validation, le nom de branche ou le nom de balise qui correspond à la version du code source à générer. Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Pour Amazon S3, ID de version de l'objet qui représente le fichier ZIP d'entrée de génération à utiliser.

Si `sourceVersion` est spécifié à la build, cette version est prioritaire sur cette `sourceVersion` (au niveau du projet). Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

artefacts

Obligatoire. Un [ProjectArtifacts](#) objet qui contient des informations sur les paramètres d'artefact de sortie de ce projet de construction. Après avoir ajouté un objet `artifacts`, vous pouvez ajouter jusqu'à 12 autres artefacts à l'aide de `addArtifact`. Il s'agit notamment des paramètres suivants :

artifacts/type

Obligatoire. Le type d'objet de sortie de la build. Les valeurs valides sont :

- CODEPIPELINE
- NO_ARTIFACTS
- S3

artifacts/ emplacement

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Nom du bucket de sortie que vous avez créé ou identifié dans les prérequis.

artifacts/ chemin

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Le chemin d'accès au compartiment de sortie où placer le fichier ou le dossier ZIP. Si vous ne spécifiez aucune valeur pour `path`, CodeBuild utilise `namespaceType` (si spécifié) et `name` pour déterminer le chemin et le nom du fichier ZIP ou du dossier de sortie de compilation. Par exemple, si vous spécifiez `MyPath` pour `path` et `MyArtifact.zip` pour `name`, le chemin et le nom seront `MyPath/MyArtifact.zip`.

artifacts/ NamespaceType

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

L'espace de noms du fichier ZIP ou du dossier de sortie de compilation. Les valeurs valides sont `BUILD_ID` et `NONE`. Utilisez `BUILD_ID` pour insérer l'ID de génération dans le chemin du fichier ZIP ou dossier de sortie de génération. Dans le cas contraire, utilisez `NONE`. Si vous ne spécifiez aucune valeur pour `namespaceType`, CodeBuild utilise `path` (si spécifié) et `name` pour déterminer le chemin et le nom du fichier ZIP ou du dossier de sortie de compilation. Par exemple, si vous spécifiez `MyPath BUILD_ID` pour `pathnamespaceType`, pour `name` et `MyArtifact.zip` pour `name`, le chemin et le nom seront `MyPath/build-ID/MyArtifact.zip`.

artefacts/nom

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Le nom du fichier ZIP ou du dossier de sortie de compilation qu'il contientlocation. Par exemple, si vous spécifiez MyPath pour path et MyArtifact.zip pourname, le chemin et le nom serontMyPath/MyArtifact.zip.

artefacts/ overrideArtifactName

Utilisé uniquement avec le type d'artefact S3. Non utilisé pour les autres types d'artefacts.

Facultatif. S'il est défini surtrue, le nom spécifié dans le artifacts bloc du fichier buildspec est remplacé. name Pour de plus amples informations, veuillez consulter [Référence de spécification de construction pour CodeBuild](#).

artefacts/emballages

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Facultatif. Spécifie comment emballer les artefacts. Les valeurs autorisées sont les suivantes :
NONE

Créez un dossier contenant les artefacts de construction. C'est la valeur par défaut.

ZIP

Créez un fichier ZIP contenant les artefacts de construction.

secondaryArtifacts

Facultatif. Tableau d'[ProjectArtifacts](#)objets contenant des informations sur les paramètres des artefacts secondaires d'un projet de construction. Vous pouvez ajouter jusqu'à 12 artefacts secondaires. secondaryArtifacts utilise un grand nombre de paramètres communs à l'objet .

cache

Obligatoire. Un [ProjectCache](#)objet qui contient des informations sur les paramètres de cache de ce projet de génération. Pour de plus amples informations, veuillez consulter [Constitutions de cache](#).

environment

Obligatoire. Un [ProjectEnvironment](#)objet qui contient des informations sur les paramètres de l'environnement de construction de ce projet. Ces paramètres sont les suivants :

environnement/type

Obligatoire. Le type d'environnement de génération. Pour plus d'informations, reportez-vous à la section [Type](#) dans la référence de CodeBuild l'API.

environnement/image

Obligatoire. Identifiant d'image Docker utilisé par cet environnement de génération.

Généralement, cet identifiant est exprimé sous la forme *image-name:tag*. Par exemple, dans le référentiel Docker CodeBuild utilisé pour gérer ses images Docker, cela pourrait être le cas. `aws/codebuild/standard:5.0` Dans Docker Hub, `maven:3.3.9-jdk-8`. Dans Amazon ECR, `account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag`. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).

environnement/ ComputeType

Obligatoire. Spécifie les ressources de calcul utilisées par cet environnement de génération. Pour plus d'informations, consultez [ComputeType](#) dans CodeBuild le Guide de référence des API.

environnement/certificat

Facultatif. L'ARN du compartiment Amazon S3, le préfixe de chemin et la clé d'objet contenant le certificat codé PEM. La clé d'objet peut être juste le fichier .pem ou un fichier .zip contenant le certificat codé PEM. Par exemple, si le nom de votre compartiment Amazon S3 est *<my-bucket>*, votre préfixe de chemin est *<cert>* et le nom de votre clé d'objet est *<certificate.pem>*, les formats acceptables pour `certificate` sont *<my-bucket/cert/certificate.pem>* ou `arn:aws:s3:::<my-bucket/cert/certificate.pem>`.

environnement/ Variables d'environnement

Facultatif. Tableau d'[EnvironmentVariable](#) objets contenant les variables d'environnement que vous souhaitez spécifier pour cet environnement de génération. Chaque variable d'environnement est exprimée sous la forme d'un name objet contenant a name, value, et type. value type

La console et AWS CLI les utilisateurs peuvent voir toutes les variables d'environnement. Si vous n'avez aucune inquiétude quant à la visibilité de votre variable d'environnement, définissez name et value et réglez type sur PLAINTEXT.

Nous vous recommandons de stocker les variables d'environnement contenant des valeurs sensibles, telles qu'un identifiant de clé d' AWS accès, une clé d'accès AWS secrète ou un mot de passe, en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store ou

AWS Secrets Manager. Pour `name`, pour ce paramètre stocké, définissez un identifiant CodeBuild à référencer.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, pour `value`, définissez le nom du paramètre tel qu'il est stocké dans le Parameter Store. Définissez `type` sur `PARAMETER_STORE`. À l'aide d'un paramètre nommé `/CodeBuild/dockerLoginPassword` titre d'exemple, définissez `name` sur `LOGIN_PASSWORD`. Définissez `value` sur `/CodeBuild/dockerLoginPassword`. Définissez `type` sur `PARAMETER_STORE`.

Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez `Create parameter` (Créer un paramètre), puis suivez les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre `/CodeBuild/` tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters`. Si vous avez sélectionné `Nouveau rôle de service` plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment `Existing service role` (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de construction fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi `Nouveau rôle de service`, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de paramètres qui ne commencent pas par `/CodeBuild/`. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres de l'espace de /CodeBuild/ noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée MY_VAR avec la valeur my_value et que vous définissez une variable d'environnement nommée MY_VAR avec la valeur other_value, la valeur my_value est remplacée par other_value. De même, si l'image Docker contient déjà une variable d'environnement nommée PATH avec la valeur /usr/local/sbin:/usr/local/bin et que vous définissez une variable d'environnement nommée PATH avec la valeur \$PATH:/usr/share/ant/bin, la valeur /usr/local/sbin:/usr/local/bin est remplacée par la valeur littérale \$PATH:/usr/share/ant/bin.

Ne définissez pas de variables d'environnement avec un nom commençant par CODEBUILD_. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration buildspec a la priorité la plus faible.

Si vous utilisez Secrets Manager, pourvalue, définissez le nom du paramètre tel qu'il est enregistré dans Secrets Manager. Définissez type sur SECRETS_MANAGER. À l'aide d'un secret nommé à /CodeBuild/dockerLoginPassword titre d'exemple, définissez name surLOGIN_PASSWORD. Définissez value sur /CodeBuild/dockerLoginPassword. Définissez type sur SECRETS_MANAGER.

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par /CodeBuild/ (par exemple,/CodeBuild/dockerLoginPassword). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser l'actionsmanager:GetSecretValueaction. Si vous avez sélectionné Nouveau rôle de

service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par /CodeBuild/, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par /CodeBuild/. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par /CodeBuild/.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de /CodeBuild/ noms dans le Gestionnaire de secrets.

Environnement/RegistryCredential

Facultatif. [RegistryCredential](#) Objet qui spécifie les informations d'identification permettant d'accéder à un registre Docker privé.

Environnement/registration/accréditation

Spécifie l'ARN ou le nom des informations d'identification créées à l'aide de AWS Managed Services. Vous pouvez utiliser le nom des informations d'identification uniquement si elles existent dans votre région actuelle.

Environnement/accréditation de registre/fournisseur d'informations d'identification

La seule valeur valide est SECRETS_MANAGER.

Lorsque cela est défini :

- `imagePullCredentials` doit être défini sur `SERVICE_ROLE`.
- L'image ne peut pas être une image sélectionnée ou une image Amazon ECR.

environnement/ Type imagePullCredentials

Facultatif. Le type d'informations d'identification CodeBuild utilisé pour extraire des images dans votre build. Deux valeurs sont valides :

CONSTRUCTION DE CODE

`CODEBUILD` indique qu'il CodeBuild utilise ses propres informations d'identification. Vous devez modifier votre politique de dépôt Amazon ECR pour faire confiance au principal du CodeBuild service.

RÔLE_SERVICE

Spécifie que CodeBuild utilise le rôle de service de votre projet de construction.

Lorsque vous utilisez une image de registre privé ou entre comptes, vous devez utiliser les informations d'identification de `SERVICE_ROLE`. Lorsque vous utilisez une image CodeBuild sélectionnée, vous devez utiliser des `CODEBUILD` informations d'identification.

environnement/ PrivilegedMode

Définissez cette valeur `true` uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre fichier `buildspec` en exécutant les commandes de génération ci-après. N'exécutez pas ces commandes si vous avez spécifié une image d'environnement de génération fournie par CodeBuild avec la prise en charge Docker.

Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux](#) Capabilities sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

serviceRole

Obligatoire. L'ARN du rôle de service est CodeBuild utilisé pour interagir avec les services pour le compte de l'utilisateur (par exemple, `arn:aws:iam::account-id:role/role-name`).

autoRetryLimit

Facultatif. Le nombre de tentatives automatiques supplémentaires après l'échec d'une compilation. Par exemple, si la limite de tentatives automatiques est définie sur 2, l'API CodeBuild appellera l'RetryBuildAPI pour réessayer automatiquement votre build jusqu'à 2 fois supplémentaires.

timeoutInMinutes

Facultatif. Le nombre de minutes, compris entre 5 et 2160 (36 heures), après lesquelles la construction CodeBuild s'arrête si elle n'est pas terminée. À défaut de précision, la valeur par défaut 60 est utilisée. Pour déterminer si et quand une compilation CodeBuild a été arrêtée en raison d'un délai d'attente, exécutez la batch-get-builds commande. Pour déterminer si la génération s'est arrêtée, recherchez dans la sortie une valeur buildStatus pour FAILED. Pour déterminer quand la génération a expiré, recherchez dans la sortie la valeur endTime associée à une valeur phaseStatus pour TIMED_OUT.

queuedTimeoutInMinutes

Facultatif. Le nombre de minutes, compris entre 5 et 480 (8 heures), après quoi la construction CodeBuild s'arrête si elle est toujours en file d'attente. À défaut de précision, la valeur par défaut 60 est utilisée.

encryptionKey

Facultatif. Alias ou ARN du fichier AWS KMS key utilisé par CodeBuild pour chiffrer la sortie de compilation. Si vous spécifiez un alias, utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID` ou, s'il existe un alias, utilisez le format `alias/key-alias`. Si elle n'est pas spécifiée, la clé KMS AWS gérée pour Amazon S3 est utilisée.

balises

Facultatif. Un tableau d'objets [Tag](#) qui fournissent les balises que vous souhaitez associer à ce projet de construction. Vous pouvez spécifier jusqu'à 50 balises. Ces balises peuvent être utilisées par n'importe quel AWS service prenant en charge les balises de projet de CodeBuild construction. Chaque balise est exprimée sous la forme d'un objet avec a key et value a.

vpcConfig

Facultatif. [VpcConfig](#)Objet contenant des informations sur la configuration VPC de votre projet. Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Ces propriétés incluent :

vpclId

Obligatoire. L'ID VPC que CodeBuild utilise. Exécutez cette commande pour obtenir une liste de tous les VPC de votre région :

```
aws ec2 describe-vpcs --region <region-ID>
```

sous-réseaux

Obligatoire. Tableau de sous-réseaux IDs qui inclut les ressources utilisées par CodeBuild. Exécutez cette commande pour les obtenir IDs :

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

securityGroupIds

Obligatoire. Tableau de groupes de sécurité IDs utilisé par CodeBuild pour autoriser l'accès aux ressources du VPC. Exécutez cette commande pour les obtenir IDs :

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

badgeEnabled

Facultatif. Spécifie s'il faut inclure des badges de construction dans votre CodeBuild projet. Définissez sur `true` pour activer la création de badges, ou `false` autre. Pour de plus amples informations, veuillez consulter [Créez un échantillon de badges avec CodeBuild](#).

Configuration des journaux

Un [LogsConfig](#) objet qui contient des informations sur l'emplacement des journaux de cette version.

Configuration des journaux/ cloudWatchLogs

[CloudWatchLogsConfig](#) Objet contenant des informations sur le transfert de CloudWatch journaux vers Logs.

LogsConfig/ S3 Logs

Un LogsConfig objet [S3](#) qui contient des informations sur le transfert de journaux vers Amazon S3.

fileSystemLocations

Facultatif. Tableau d'[ProjectFileSystemsLocation](#) objets contenant des informations sur votre configuration Amazon EFS.

buildBatchConfig

Facultatif. L'buildBatchConfig objet est une [ProjectBuildBatchConfig](#) structure qui contient les informations de configuration de génération par lots pour le projet.

buildBatchConfig/ServiceRole

L'ARN du rôle de service pour le projet de génération par lots.

buildBatchConfig/Combinez des artefacts

Valeur booléenne qui indique s'il faut combiner les artefacts de construction pour la construction par lots dans un seul emplacement d'artefact.

buildBatchConfig/restrictions/ maximumBuildsAllowed

Le nombre maximum de builds autorisés.

buildBatchConfig/restrictions/ computeTypesAllowed

Tableau de chaînes qui spécifient les types de calcul autorisés pour la construction par lots.

Reportez-vous [à la section Types de calcul de l'environnement](#) de génération pour ces valeurs.

buildBatchConfig/restrictions/ Flottes autorisées

Tableau de chaînes qui spécifient les flottes autorisées pour la génération par lots. Voir [Exécuter des builds sur des flottes de capacité réservée](#) pour plus d'informations.

buildBatchConfig/timeoutInMinutes

Durée maximale, en minutes, pendant laquelle la génération par lots doit être terminée.

buildBatchConfig/batchReportMode

Spécifie comment les rapports d'état de compilation sont envoyés au fournisseur source pour la compilation par lots. Les valeurs valides sont les suivantes :

REPORT_AGGREGATED_BATCH

(Par défaut) Regrouper tous les états de compilation dans un seul rapport d'état.

REPORT_INDIVIDUAL_BUILDS

Envoyer un rapport d'état distinct pour chaque compilation individuelle.

concurrentBuildLimit

Nombre maximal de générations simultanées autorisées pour ce projet.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Création du projet

Pour créer le projet, réexécutez la [create-project](#) commande en passant votre fichier JSON :

```
aws codebuild create-project --cli-input-json file://<json-file>
```

En cas de succès, la représentation JSON d'un objet [Project](#) apparaît dans la sortie de la console. Voir la [syntaxe de CreateProject réponse](#) pour un exemple de ces données.

A l'exception du nom de projet de génération, vous pouvez modifier les paramètres du projet de génération ultérieurement. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).

Pour démarrer l'exécution d'une génération, consultez [Exécution d'une génération \(AWS CLI\)](#).

Si votre code source est stocké dans un GitHub référentiel et que vous CodeBuild souhaitez le reconstruire chaque fois qu'une modification de code est envoyée au référentiel, consultez [Lancement automatique de l'exécution de générations \(AWS CLI\)](#).

Création d'un projet de génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Création d'un projet de génération (CloudFormation)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec CloudFormation, consultez [le CloudFormation modèle figurant CodeBuild](#) dans le guide de AWS CloudFormation l'utilisateur.

Création d'une règle de notification

Vous pouvez utiliser des règles de notification pour informer les utilisateurs de changements importants, notamment des succès et des échecs de création. Les règles de notification spécifient à la fois les événements et le sujet Amazon SNS utilisé pour envoyer des notifications. Pour plus d'informations, consultez [Que sont les notifications ?](#)

Vous pouvez utiliser la console ou le AWS CLI pour créer des règles de notification pour AWS CodeBuild.

Pour créer une règle de notification (console)

1. Connectez-vous à la CodeBuild console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Choisissez Build (Générer), puis Build projects (Générer des projets), puis sélectionnez un projet de création auquel vous souhaitez ajouter des notifications.
3. Sur la page du projet de création, choisissez Notify (Notifier), puis Create notification rule (Créer une règle de notification). Vous pouvez également accéder à la page Settings (Paramètres) du projet de génération et choisir Create notification rule (Créer une règle de notification).
4. Dans Notification name (Nom de la notification), saisissez le nom de la règle.
5. Dans Type de détail, choisissez Basic si vous souhaitez que seules les informations fournies à Amazon soient EventBridge incluses dans la notification. Choisissez Complet si vous souhaitez inclure les informations fournies à Amazon EventBridge et les informations susceptibles d'être fournies par le CodeBuild ou le gestionnaire de notifications.

Pour plus d'informations, consultez [Présentation du contenu des notifications et de la sécurité](#).

6. Dans Événements qui déclenchent des notifications, sélectionnez les événements pour lesquels vous souhaitez envoyer des notifications. Pour de plus amples informations, veuillez consulter [Événements pour les règles de notification sur les projets de génération](#).
7. Dans Targets (Cibles), effectuez l'une des actions suivantes :

- Si vous avez déjà configuré une ressource à utiliser avec les notifications, dans Choisir le type de cible, sélectionnez Amazon Q Developer dans les applications de chat (Slack) ou le sujet SNS. Dans Choisir une cible, choisissez le nom du client (pour un client Slack configuré dans Amazon Q Developer dans les applications de chat) ou le nom de ressource Amazon (ARN) du sujet Amazon SNS (pour les sujets Amazon SNS déjà configurés avec la politique requise pour les notifications).
- Si vous n'avez pas configuré de ressource à utiliser avec les notifications, choisissez Create target (Créer une cible), puis SNS topic (Rubrique SNS). Donnez un nom à la rubrique après codestar-notifications-, puis choisissez Create (Créer).

Note

- Si vous créez la rubrique Amazon SNS dans le cadre de la création de la règle de notification, la stratégie qui permet à la fonctionnalité de notifications de publier des événements dans la rubrique est appliquée automatiquement. L'utilisation d'une rubrique créée pour les règles de notification vous permet de vous assurer que vous n'abonnez que les utilisateurs qui doivent recevoir des notifications sur ce référentiel.
- Vous ne pouvez pas créer un client Amazon Q Developer dans les applications de chat dans le cadre de la création d'une règle de notification. Si vous choisissez Amazon Q Developer dans les applications de chat (Slack), vous verrez un bouton vous demandant de configurer un client dans Amazon Q Developer dans les applications de chat. Le choix de cette option ouvre la console Amazon Q Developer dans les applications de chat. Pour plus d'informations, consultez [Configurer les intégrations entre les notifications et Amazon Q Developer dans les applications de chat](#).
- Si vous souhaitez utiliser une rubrique Amazon SNS existante comme cible, vous devez ajouter la politique requise pour les AWS CodeStar notifications en plus de toute autre politique susceptible d'exister pour cette rubrique. Pour de plus amples informations, veuillez consulter [Configurer les rubriques Amazon SNS existantes pour les notifications](#) et [Présentation du contenu des notifications et de la sécurité](#).

8. Pour terminer la création de la règle, choisissez Submit (Soumettre).
9. Vous devez inscrire les utilisateurs à la rubrique Amazon SNS relative à la règle pour qu'ils puissent recevoir des notifications. Pour plus d'informations, consultez [Abonner les utilisateurs](#)

[aux rubriques Amazon SNS qui sont](#) des cibles. Vous pouvez également configurer l'intégration entre les notifications et Amazon Q Developer dans les applications de chat afin d'envoyer des notifications aux forums de discussion Amazon Chime. Pour plus d'informations, consultez [Configurer l'intégration entre les notifications et Amazon Q Developer dans les applications de chat](#).

Pour créer une règle de notification (AWS CLI)

1. Dans un terminal ou une invite de commandes, exécutez la commande `create-notification-rule` pour générer le squelette JSON :

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

Vous pouvez donner au fichier le nom de votre choix. Dans cet exemple, le fichier est nommé *rule.json*.

2. Ouvrez le fichier JSON dans un éditeur de texte brut et modifiez-le pour y inclure la ressource, les types d'événements et la cible que vous souhaitez pour la règle. L'exemple suivant montre une règle de notification nommée **MyNotificationRule** pour un projet de construction nommé *MyBuildProject* dans un AWS compte avec cet ID *123456789012*. Les notifications sont envoyées avec le type de détail complet à une rubrique Amazon SNS nommée *codestar-notifications-MyNotificationTopic* when builds are successful :

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeIds": [  
    "codebuild-project-build-state-succeeded"  
  ],  
  "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",  
  "Targets": [  
    {  
      "TargetType": "SNS",  
      "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-  
notifications-MyNotificationTopic"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL"  
}
```

Enregistrez le fichier.

3. À l'aide du fichier que vous venez de modifier, à partir du terminal ou de la ligne de commande, exécutez à nouveau la commande `create-notification-rule` pour créer la règle de notification :

```
aws codestarnotifications create-notification-rule --cli-input-json
  file://rule.json
```

4. En cas de réussite, la commande renvoie l'ARN de la règle de notification, comme suit :

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Modifier les paramètres du projet de construction dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs pour modifier les paramètres d'un projet de construction. AWS CLI

Si vous ajoutez des rapports de test à un projet de génération, assurez-vous que votre rôle IAM dispose des autorisations décrites dans [Autorisations relatives aux rapports de test](#).

Rubriques

- [Modification des paramètres d'un projet de génération \(console\)](#)
- [Modification des paramètres d'un projet de génération \(AWS CLI\)](#)
- [Modification des paramètres d'un projet de génération \(AWS SDKs\)](#)

Modification des paramètres d'un projet de génération (console)

Pour modifier les paramètres d'un projet de génération, effectuez la procédure suivante :

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :

- Cliquez sur le lien du projet de génération à modifier, puis choisissez Détails de génération.
- Cliquez sur le bouton en regard du projet de génération à modifier, choisissez View details (Afficher les détails), puis Détails de génération.

Vous pouvez modifier les sections suivantes :

Sections

- [Configuration du projet](#)
- [Source](#)
- [Environnement](#)
- [Spécifications de construction](#)
- [Configuration par lots](#)
- [Artefacts](#)
- [Journaux](#)

Configuration du projet

Dans la section Configuration du projet, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes.

Description

Entrez une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

Construire un badge

Sélectionnez Enable build badge (Activer le badge de génération) pour rendre le statut de génération de votre projet visible et intégrable. Pour de plus amples informations, veuillez consulter [Exemple de badges de génération](#).

Note

Le badge de construction ne s'applique pas si votre fournisseur source est Amazon S3.

Activer la limite de génération simultanée

Si vous souhaitez limiter le nombre de builds simultanés pour ce projet, effectuez les opérations suivantes :

1. Sélectionnez Restreindre le nombre de versions simultanées que ce projet peut démarrer.
2. Dans Limite de génération simultanée, entrez le nombre maximum de versions simultanées autorisées pour ce projet. Cette limite ne peut pas être supérieure à la limite de création simultanée définie pour le compte. Si vous essayez de saisir un nombre supérieur à la limite du compte, un message d'erreur s'affiche.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Permettre l'accès public aux builds

Pour que les résultats de génération de votre projet soient accessibles au public, y compris aux utilisateurs n'ayant pas accès à un AWS compte, sélectionnez Activer l'accès public aux versions et confirmez que vous souhaitez rendre les résultats de génération publics. Les propriétés suivantes sont utilisées pour les projets de construction publics :

Rôle du service public de construction

Sélectionnez Nouveau rôle de service si vous souhaitez CodeBuild créer un nouveau rôle de service pour vous, ou Rôle de service existant si vous souhaitez utiliser un rôle de service existant.

Le rôle de service de construction public CodeBuild permet de lire les CloudWatch journaux et de télécharger les artefacts Amazon S3 pour les versions du projet. Cela est nécessaire pour mettre les journaux de construction et les artefacts du projet à la disposition du public.

Rôle de service

Entrez le nom du nouveau rôle de service ou d'un rôle de service existant.

Pour que les résultats de compilation de votre projet soient privés, décochez Activer l'accès public aux builds.

Pour de plus amples informations, veuillez consulter [Obtenir un projet de construction public URLs](#).

Warning

Les points suivants doivent être pris en compte lorsque vous publiez les résultats de construction de votre projet :

- Tous les résultats de construction, les journaux et les artefacts d'un projet, y compris les versions exécutées lorsque le projet était privé, sont accessibles au public.
- Tous les journaux de construction et les artefacts sont accessibles au public. Les variables d'environnement, le code source et d'autres informations sensibles peuvent avoir été générés dans les journaux de construction et les artefacts. Vous devez faire attention aux informations qui sont affichées dans les journaux de construction. Voici certaines des meilleures pratiques :
 - Ne stockez pas de valeurs sensibles, en particulier les identifiants de clé AWS d'accès et les clés d'accès secrètes, dans les variables d'environnement. Nous vous recommandons d'utiliser un magasin de paramètres Amazon EC2 Systems Manager AWS Secrets Manager ou de stocker des valeurs sensibles.
 - Suivez cette [Bonnes pratiques d'utilisation des webhooks](#) procédure pour limiter les entités qui peuvent déclencher une construction, et ne stockez pas la spécification de construction dans le projet lui-même, afin de garantir que vos webhooks sont aussi sécurisés que possible.
- Un utilisateur malveillant peut utiliser des versions publiques pour distribuer des artefacts malveillants. Nous recommandons aux administrateurs de projet de passer en revue toutes les pull requests afin de vérifier qu'il s'agit d'une modification légitime. Nous vous recommandons également de valider tous les artefacts à l'aide de leurs checksums afin de vous assurer que les bons artefacts sont téléchargés.

Informations supplémentaires

Pour les balises, entrez le nom et la valeur de toutes les balises que vous souhaitez que les AWS services d'assistance utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.

Source

Dans la section Source, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Fournisseur de source

Choisissez le type de fournisseur de code source. Utilisez les listes suivantes pour effectuer des sélections adaptées à votre fournisseur de source :

Note

CodeBuild ne prend pas en charge Bitbucket Server.

Amazon S3

Compartiment

Choisissez le nom du compartiment d'entrée contenant le code source.

Clé d'objet S3 ou dossier S3

Entrez le nom du fichier ZIP ou le chemin d'accès au dossier contenant le code source. Entrez une barre oblique (/) pour tout télécharger dans le compartiment S3.

Version de la source

Entrez l'ID de version de l'objet qui représente la version de votre fichier d'entrée. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

CodeCommit

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Type de référence

Choisissez Branch, Git tag ou Commit ID pour spécifier la version de votre code source. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez de créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Bitbucket

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez CodeConnectionsle OAuthmot de passe de l'application ou le jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild.

Connexion

Sélectionnez une connexion Bitbucket ou un secret de Secrets Manager pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository dans mon compte Bitbucket ou Public repository et saisissez l'URL du référentiel.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le name paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

Pour l'URL cible, entrez la valeur à utiliser pour le url paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [Événements du webhook Bitbucket](#).

GitHub

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez GitHub Application ou jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild. OAuth

Connexion

Sélectionnez une GitHub connexion ou un secret du Gestionnaire de Secrets pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository in my GitHub account, Public repository ou GitHub Scoped Webhook et entrez l'URL du référentiel.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, consultez [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le context paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour URL cible, entrez la valeur à utiliser pour le target_url paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitHub Enterprise Server

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

Choisissez CodeConnections un jeton d'accès personnel auquel vous souhaitez vous connecter CodeBuild.

Connexion

Sélectionnez une connexion GitHub Enterprise ou un secret Secrets Manager pour vous connecter via le type de connexion que vous avez spécifié.

Référentiel

Choisissez Repository in my GitHub Enterprise account ou GitHub Enterprise Scoped Webhook et entrez l'URL du référentiel.

Version de la source

Entrez une pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le context paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour URL cible, entrez la valeur à utiliser pour le target_url paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

SSL non sécurisé

Sélectionnez Activer le protocole SSL non sécurisé pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub d'entreprise.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitLab

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

CodeConnectionsest utilisé pour se connecter GitLab à CodeBuild.

Connexion

Sélectionnez une GitLab connexion par laquelle vous souhaitez vous connecter CodeConnections.

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Version de la source

Entrez un identifiant de pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

GitLab Self Managed

Accréditation

Choisissez Informations d'identification source par défaut ou Informations d'identification source personnalisées et suivez les instructions pour gérer les informations d'identification source par défaut ou personnaliser les informations d'identification source.

Type de connexion

CodeConnection est utilisé pour connecter GitLab Self Managed à CodeBuild.

Connexion

Sélectionnez une connexion GitLab autogérée par laquelle vous souhaitez vous connecter CodeConnections.

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Version de la source

Entrez un identifiant de pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Environnement

Dans la section Environnement, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Modèle de provisionnement

Pour modifier le modèle de provisionnement, choisissez Modifier le modèle de provisionnement et effectuez l'une des opérations suivantes :

- Pour utiliser des flottes à la demande gérées par AWS CodeBuild, choisissez On-Demand. Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites à la fin de la construction. Les flottes à la demande sont entièrement

gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.

- Pour utiliser des flottes de capacité réservée gérées par AWS CodeBuild, choisissez Capacité réservée, puis sélectionnez un nom de flotte. Avec les flottes de capacité réservée, vous configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

Pour plus d'informations, consultez [Exécutez des builds sur des flottes à capacité réservée](#).

Image de l'environnement

Pour modifier l'image de construction, choisissez Remplacer l'image et effectuez l'une des opérations suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Rôle de service

Effectuez l'une des actions suivantes :

- Si vous n'avez pas de rôle CodeBuild de service, choisissez Nouveau rôle de service. Dans Role name, entrez un nom pour le nouveau rôle.
- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

Note

Lorsque vous utilisez la console pour créer un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

Configuration supplémentaire

Expiration

Spécifiez une valeur, comprise entre 5 minutes et 36 heures, après quoi la CodeBuild génération s'arrête si elle n'est pas terminée. Si les valeurs de heures et minutes sont laissées vides, la valeur par défaut de 60 minutes est utilisée.

privilégié

Sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos versions bénéficient de privilèges élevés. uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre spécification de génération en exécutant les commandes de génération ci-après. N'exécutez pas ces commandes si vous avez choisi une image d'environnement de construction fournie CodeBuild par le support Docker.

 Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux](#) Capabilities sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources du VPCs

Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Calcul

Choisissez l'une des options disponibles.

Informations d'identification du registre

Spécifiez un identifiant de registre lorsque le projet est configuré avec une image de registre non privée.

 Note

Ces informations d'identification ne seront utilisées que si les images sont remplacées par celles provenant de registres privés.

Variables d'environnement

Entrez le nom et la valeur, puis choisissez le type de chaque variable d'environnement à utiliser pour les builds.

Note

CodeBuild définit automatiquement la variable d'environnement pour votre AWS région. Vous devez définir les variables d'environnement suivantes si vous ne les avez pas ajoutées dans votre fichier `buildspec.yml` :

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

La console et AWS CLI les utilisateurs peuvent voir les variables d'environnement. Si la visibilité de vos variables d'environnement ne vous pose pas de problème, définissez les zones Nom et Valeur, puis définissez Type sur Texte brut.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un identifiant de clé d' AWS accès, une clé d'accès AWS secrète ou un mot de passe en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager ou.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, choisissez Parameter dans Type. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value, entrez le nom du paramètre tel qu'il est stocké dans le magasin de paramètres Amazon EC2 Systems Manager. Si l'on prend comme exemple un paramètre nommé `/CodeBuild/dockerLoginPassword`, pour Type, choisissez Parameter (Paramètre). Pour Nom, saisissez `LOGIN_PASSWORD`. Pour le champ Valeur, saisissez `/CodeBuild/dockerLoginPassword`.

Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant `/CodeBuild/` par (par exemple `/CodeBuild/dockerLoginPassword`). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez Create parameter (Créer un paramètre), puis suivez

les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre `/CodeBuild/` tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters` l'action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de paramètres qui ne commencent pas par `/CodeBuild/`. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres sous l'espace de `/CodeBuild/` noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par CODEBUILD_. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration buildspec a la priorité la plus faible.

Si vous utilisez Secrets Manager, pour Type, choisissez Secrets Manager. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value (Valeur), saisissez un reference-key à l'aide du modèle *secret-id:json-key:version-stage:version-id*. Pour plus d'informations, consultez [Secrets Manager reference-key in the buildspec file](#).

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par /CodeBuild/ (par exemple, /CodeBuild/dockerLoginPassword). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser l'actionsmanager:GetSecretValueaction. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par /CodeBuild/, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par /CodeBuild/. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par /CodeBuild/.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de /CodeBuild/ noms dans le Gestionnaire de secrets.

Spécifications de construction

Dans la section Buildspec, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Spécifications de construction

Effectuez l'une des actions suivantes :

- Si votre code source inclut un fichier buildspec, choisissez Utiliser un fichier buildspec. Par défaut, CodeBuild recherche un fichier nommé `buildspec.yml` dans le répertoire racine de code source. Si votre fichier buildspec utilise un nom ou un emplacement différent, entrez son chemin depuis la racine source dans le nom Buildspec (par exemple, `ou.buildspec-two.yml configuration/buildspec.yml`). Si le fichier buildspec se trouve dans un compartiment S3, il doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier buildspec à l'aide de son ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Si votre code source ne comprend pas de fichier de spécification de génération ou si vous souhaitez exécuter des commandes de génération différentes de celles spécifiées pour la phase build dans le fichier `buildspec.yml` au sein du répertoire racine du code source, choisissez Insérer des commandes de génération. Pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase build. Pour plusieurs commandes, séparez celles-ci avec `&&` (par exemple, `mvn test && mvn package`). Pour exécuter des commandes dans d'autres phases, ou si vous avez une longue liste de commandes pour la build phase, ajoutez un `buildspec.yml` fichier dans le répertoire racine du code source, ajoutez les commandes au fichier, puis choisissez Utiliser le fichier `buildspec.yml` dans le répertoire racine du code source.

Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

Configuration par lots

Dans la section Configuration par lots, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration. Pour de plus amples informations, veuillez consulter [Exécuter des builds par lots](#).

Vous pouvez modifier les propriétés suivantes :

Rôle du service Batch

Fournit le rôle de service pour les compilations par lots.

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle de service par lots, choisissez Nouveau rôle de service. Dans Rôle de service, entrez le nom du nouveau rôle.
- Si vous avez un rôle de service par lots, choisissez Rôle de service existant. Dans Rôle de service, choisissez le rôle de service.

Les générations en lot introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est obligatoire car CodeBuild il doit être capable d'appeler les RetryBuild actions StartBuildStopBuild, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même rôle qu'ils utilisent dans leur build, pour deux raisons :

- Donner au rôle de build StartBuild, StopBuild et RetryBuild les autorisations permettrait à une seule build de lancer davantage de builds via le buildspec.
- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds elles-mêmes puissent contourner ces restrictions.

Types de calcul autorisés pour le traitement par lots

Sélectionnez les types de calcul autorisés pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Flottes autorisées pour le lot

Sélectionnez les flottes autorisées pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Nombre maximal de builds autorisés par lot

Entrez le nombre maximum de builds autorisés dans le lot. Si un lot dépasse cette limite, il échouera.

Délai d'expiration du Batch

Entrez la durée maximale pendant laquelle la génération par lots doit être terminée.

Combinez des artefacts

Sélectionnez Combiner tous les artefacts du lot en un seul emplacement pour que tous les artefacts du lot soient combinés en un seul emplacement.

Mode de rapport par lots

Sélectionnez le mode de rapport d'état de construction souhaité pour les versions par lots.

Note

Ce champ n'est disponible que lorsque la source du projet est Bitbucket ou GitHub Enterprise GitHub, et l'option Signaler les statuts de construction au fournisseur de source lorsque le début et la fin de vos builds sont sélectionnés sous Source.

Constructions agrégées

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient combinés dans un seul rapport d'état.

Constructions individuelles

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient signalés séparément.

Artefacts

Dans la section Artefacts, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Type

Effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer des artefacts de sortie de génération, choisissez Aucun artefact. Vous pouvez le faire si vous exécutez uniquement des tests de compilation ou si vous souhaitez transférer une image Docker vers un référentiel Amazon ECR.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. (Si vous souhaitez produire un fichier ZIP et que vous voulez que celui-ci ait une extension de fichier, veillez à l'inclure après le nom de fichier ZIP.)
 - Sélectionnez Activer la gestion sémantique des versions si vous voulez qu'un nom spécifié dans le fichier buildspec remplace le nom spécifié dans la console. Le nom figurant dans un fichier buildspec est calculé au moment de la génération et utilise le langage de commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).
 - Si vous ne souhaitez pas que vos artefacts de génération soient chiffrés, choisissez Remove artifacts encryption (Supprimer le chiffrement des artefacts).

Pour chaque ensemble d'artefacts secondaire que vous souhaitez :

1. Pour Artifact identifier (Identifiant d'artefact), saisissez une valeur de moins de 128 caractères et contenant uniquement des caractères alphanumériques et des traits de soulignement.
2. Choisissez Add artifact (Ajouter un artefact).
3. Suivez les étapes précédentes pour configurer vos artefacts secondaires.
4. Choisissez Save artifact (Enregistrer l'artefact).

Configuration supplémentaire

Clé de chiffrement

Effectuez l'une des actions suivantes :

- Pour utiliser la Clé gérée par AWS Amazon S3 dans votre compte afin de chiffrer les artefacts de sortie du build, laissez la clé de chiffrement vide. Il s'agit de l'option par défaut.
- Pour utiliser une clé gérée par le client pour chiffrer les artefacts de sortie de génération, dans Clé de chiffrement, entrez l'ARN de la clé gérée par le client. Utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Type de cache

Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :

- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées

d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Des mises en cache pour améliorer les performances](#).

Journaux

Dans la section Journaux, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Choisissez les journaux que vous souhaitez créer. Vous pouvez créer des CloudWatch journaux Amazon Logs, des journaux Amazon S3 ou les deux.

CloudWatch

Si vous souhaitez obtenir CloudWatch les journaux Amazon Logs :

CloudWatch journaux

Sélectionnez Journaux CloudWatch .

Nom du groupe

Entrez le nom de votre groupe de CloudWatch journaux Amazon Logs.

Nom du stream

Entrez le nom de votre flux de journal Amazon CloudWatch Logs.

S3

Si vous souhaitez obtenir les journaux Amazon S3 :

Journaux S3

Sélectionnez Journaux S3.

Compartiment

Choisissez le nom du compartiment S3 pour vos journaux.

Préfixe de chemin

Entrez le préfixe de vos journaux.

Désactiver le chiffrement des journaux S3

Sélectionnez si vous ne voulez pas que vos journaux S3 soient chiffrés.

Modification des paramètres d'un projet de génération (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Pour mettre à jour un CodeBuild projet avec le AWS CLI, vous créez un fichier JSON avec les propriétés mises à jour et vous transmettez ce fichier à la [update-project](#) commande. Toutes les propriétés qui ne figurent pas dans le fichier de mise à jour restent inchangées.

Dans le fichier JSON de mise à jour, seules la name propriété et les propriétés modifiées sont requises. La name propriété identifie le projet à modifier. Pour toute structure modifiée, les paramètres requis pour ces structures doivent également être inclus. Par exemple, pour modifier l'environnement du projet, les environment/computeType propriétés environment/type et sont requises. Voici un exemple qui met à jour l'image de l'environnement :

```
{
  "name": "<project-name>",
  "environment": {
    "type": "LINUX_CONTAINER",
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:4.0"
  }
}
```

Si vous devez obtenir les valeurs de propriétés actuelles d'un projet, utilisez la [batch-get-projects](#) commande pour obtenir les propriétés actuelles du projet que vous modifiez et écrivez le résultat dans un fichier.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

Le *project-info.json* fichier contient un tableau de projets, il ne peut donc pas être utilisé directement pour mettre à jour un projet. Vous pouvez toutefois copier les propriétés que vous souhaitez modifier à partir du *project-info.json* fichier et les coller dans votre fichier de mise à jour comme référence pour les propriétés que vous souhaitez modifier. Pour de plus amples informations, veuillez consulter [Affichage des détails d'un projet de génération \(AWS CLI\)](#).

Modifiez le fichier JSON de mise à jour comme décrit dans [Création d'un projet de génération \(AWS CLI\)](#), et enregistrez vos résultats. Lorsque vous avez terminé de modifier le fichier JSON de mise à jour, exécutez la [update-project](#) commande en transmettant le fichier JSON de mise à jour.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

En cas de succès, le JSON du projet mis à jour apparaît dans la sortie. Si des paramètres obligatoires sont manquants, un message d'erreur s'affiche dans la sortie pour identifier les paramètres manquants. Par exemple, voici le message d'erreur qui s'affiche si le `environment/` type paramètre est absent :

```
aws codebuild update-project --cli-input-json file://update-project.json
```

```
Parameter validation failed:  
Missing required parameter in environment: "type"
```

Modification des paramètres d'un projet de génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Plusieurs jetons d'accès dans CodeBuild

CodeBuild permet de fournir des jetons d'accès à des fournisseurs tiers à partir de vos secrets dans AWS Secrets Manager ou via AWS CodeConnections des connexions. Vous pouvez définir votre secret ou votre connexion comme identifiant par défaut pour les interactions avec un fournisseur tiers spécifique tel que GitHub GitHub Enterprise ou Bitbucket.

Vous pouvez définir vos informations d'identification source à trois niveaux différents :

1. Informations d'identification au niveau du compte pour tous les projets : il s'agit des informations d'identification par défaut pour tous les projets d'un AWS compte. Ils seront utilisés dans un projet lorsqu'aucune information d'identification au niveau du projet ou de la source n'est spécifiée.
2. Informations d'identification au niveau de la source pour un référentiel spécifique : cela se produit lorsqu'un secret ou une CodeConnections connexion Secrets Manager est défini sur une source de projet. Ces informations d'identification ne seront utilisées que pour les opérations sur le référentiel source spécifié. Cela vous permet de configurer plusieurs jetons d'accès avec

des étendues d'autorisation différentes dans le même projet, sans utiliser les informations d'identification par défaut au niveau du compte.

3. Informations d'identification de secours au niveau du projet : vous pouvez définir des informations d'identification de secours au niveau du projet en l'utilisant NO_SOURCE comme type de source principale et en y définissant un secret ou une connexion. Cela peut être utilisé lorsque vous avez plusieurs sources sur un projet, mais que vous souhaitez utiliser les mêmes informations d'identification pour celles-ci, ou lorsque vous ne souhaitez pas utiliser les informations d'identification par défaut au niveau du compte pour votre projet.

Rubriques

- [Étape 1 : Création d'un secret ou d'une CodeConnections connexion dans le Gestionnaire de Secrets](#)
- [Étape 2 : Accorder au rôle IAM CodeBuild du projet l'accès aux secrets de Secrets Manager](#)
- [Étape 3 : Configuration du Gestionnaire de Secrets Manager ou CodeConnections des jetons](#)
- [Options de configuration supplémentaires](#)

Étape 1 : Création d'un secret ou d'une CodeConnections connexion dans le Gestionnaire de Secrets

Suivez les instructions suivantes pour créer un secret ou une CodeConnections connexion dans le Gestionnaire de Secrets :

- [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets.](#)
- [Créez une connexion avec GitHub](#)
- [Création d'une connexion à GitHub Enterprise Server](#)
- [Création d'une connexion à Bitbucket](#)

Étape 2 : Accorder au rôle IAM CodeBuild du projet l'accès aux secrets de Secrets Manager

Note

Avant de continuer, vous devez avoir accès au jeton créé dans Secrets Manager ou CodeConnections.

Pour accorder l'accès au rôle IAM du CodeBuild projet à Secrets Manager CodeConnections, vous devez ajouter la politique IAM suivante.

Pour accorder l'accès CodeBuild au rôle IAM du projet

1. Créez un rôle IAM pour votre CodeBuild projet en suivant les instructions correspondant [CodeBuild Autoriser l'interaction avec d'autres AWS services](#) à votre CodeBuild projet.
2. Effectuez l'une des actions suivantes :
 - Ajoutez la politique IAM suivante à votre rôle dans le CodeBuild projet pour autoriser l'accès à votre secret.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:iam::*:role/Service*"
      ]
    }
  ]
}
```

(Facultatif) Si vous utilisez des clés gérées par le AWS KMS client pour chiffrer un secret de Secrets Manager, vous pouvez ajouter la déclaration de politique suivante pour autoriser l'accès.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:iam::*:role/Service*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:SecretARN":
            "arn:aws:iam::*:role/Service*"
        }
      }
    }
  ]
}
```

- Ajoutez la politique IAM suivante à votre rôle de CodeBuild projet pour autoriser l'accès à votre connexion.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codeconnections:GetConnectionToken",
        "codeconnections:GetConnection"
      ],
    }
  ]
}
```

```
        "Resource": [
            "arn:aws:iam::*:role/Service*"
        ]
    }
}
```

Étape 3 : Configuration du Gestionnaire de Secrets Manager ou CodeConnections des jetons

Vous pouvez définir vos informations d'identification source à trois niveaux différents à l'aide de Secrets Manager ou de CodeConnections jetons.

Configurer Secrets Manager ou les CodeConnections jetons comme informations d'identification au niveau du compte

Vous pouvez configurer un secret ou une CodeConnections connexion Secrets Manager comme identifiant au niveau du compte et l'utiliser dans un projet.

AWS Management Console

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS Management Console

1. Dans le champ Source provider, choisissez Bitbucket ou GitHub Enterprise. GitHub
2. Pour Credential, effectuez l'une des opérations suivantes :
 - Choisissez Informations d'identification source par défaut pour utiliser les informations d'identification source par défaut de votre compte pour les appliquer à tous les projets.
 - a. Si vous n'êtes pas connecté à votre fournisseur de source, choisissez Gérer les informations d'identification de source par défaut.
 - b. Pour le type d'identification, choisissez un type d'identification.
 - c. Si vous le souhaitez CodeConnections, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.

Si vous avez choisi un autre type d'identifiant, dans Service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :

- Si vous avez choisi d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou de créer un nouveau secret et choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
- Si vous avez choisi de l'utiliser CodeBuild, entrez votre jeton ou votre nom d'utilisateur et le mot de passe de l'application, puis choisissez Enregistrer.
- Choisissez Identifiant source personnalisé pour utiliser un identifiant source personnalisé afin de remplacer les paramètres par défaut de votre compte.
 - a. Pour le type d'identification, choisissez un type d'identification.
 - b. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.

AWS CLI

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS CLI

- Ouvrez une invite de terminal (Linux, macOS ou Unix) ou de commande (Windows). Utilisez le AWS CLI pour exécuter la `import-source-credentials` commande.

Utilisez la commande suivante pour configurer un secret du Gestionnaire de Secrets :

```
aws codebuild import-source-credentials \  
  --token "<secret-arn>" \  
  --server-type <source-provider> \  
  --auth-type SECRETS_MANAGER \  
  --region <aws-region>
```

Utilisez la commande suivante pour configurer une CodeConnections connexion :

```
aws codebuild import-source-credentials \  
  --token "<connection-arn>" \  
  --server-type <source-provider> \  
  --auth-type CODECONNECTIONS \  
  --region <aws-region>
```

Cette commande vous permet d'importer un jeton comme identifiant de source par défaut au niveau du compte. Lorsque vous importez un identifiant à l'aide de

l'[ImportSourceCredentials](#) API, vous CodeBuild utiliserez le jeton pour toutes les interactions avec le fournisseur source, telles que les webhooks, les rapports sur l'état des builds et les opérations de clonage git, sauf si un ensemble d'informations d'identification plus spécifique a été configuré dans le projet.

Vous pouvez désormais utiliser le jeton dans votre projet de build et l'exécuter. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) et [Exécuter AWS CodeBuild les builds manuellement](#).

Configurer plusieurs jetons comme informations d'identification au niveau de la source

Pour utiliser les secrets ou les CodeConnections connexions de Secrets Manager comme informations d'identification au niveau de la source, référez directement le jeton dans le CodeBuild projet et lancez une compilation.

AWS Management Console

Pour configurer plusieurs jetons comme informations d'identification au niveau de la source dans AWS Management Console

1. Dans le champ Source provider, sélectionnez GitHub.
2. Pour Credential, effectuez l'une des opérations suivantes :
 - Choisissez Informations d'identification source par défaut pour utiliser les informations d'identification source par défaut de votre compte pour les appliquer à tous les projets.
 - a. Si vous n'êtes pas connecté à GitHub, choisissez Gérer les informations d'identification de source par défaut.
 - b. Pour le type d'identifiant, choisissez GitHub App.
 - c. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.
 - Choisissez Identifiant source personnalisé pour utiliser un identifiant source personnalisé afin de remplacer les paramètres par défaut de votre compte.
 - a. Pour le type d'identifiant, choisissez GitHub App.
 - b. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.

3. Choisissez Ajouter une source et répétez le processus de sélection de votre fournisseur source et de vos informations d'identification.

AWS CLI

Pour configurer plusieurs jetons comme informations d'identification au niveau de la source dans AWS CLI

- Ouvrez une invite de terminal (Linux, macOS ou Unix) ou de commande (Windows). Utilisez le AWS CLI pour exécuter la create-project commande.

Utilisez la commande suivante :

```
aws codebuild create-project --region <aws-region> \
  --name <project-name> \
  --artifacts type=NO_ARTIFACTS \
  --environment "type=LINUX_CONTAINER,
                  computeType=BUILD_GENERAL1_SMALL,
                  image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
  --service-role <service-role-name> \
  --source "type=GITHUB,
            location=<github-repository-1>,
            auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-1>}" \
  --secondary-sources "type=GITHUB,
                       location=<github-repository-2>,
                       auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-2>},
                       sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project-name>
```

Définissez une solution de secours pour les informations d'identification source au niveau du projet

Pour configurer le remplacement des informations d'identification de source au niveau du projet, utilisez-le NO_SOURCE comme source principale de votre projet et référez le jeton.

```
aws codebuild create-project \
  --name <project-name> \
  --service-role <service-role-name> \
```

```
--artifacts type=NO_ARTIFACTS \
--environment "type=LINUX_CONTAINER,
               computeType=BUILD_GENERAL1_SMALL,
               image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
--service-role <service-role-name> \
--source "type=NO_SOURCE,
         auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn>},
         buildspec=<buildspec>"
--secondary-sources "type=GITHUB,
                    location=<github-repository>,
                    sourceIdentifier=secondary"
```

```
aws codebuild start-build --region <aws-region> --project-name <project_name>
```

[Lors de l'utilisation NO_SOURCE, une spécification de construction est généralement fournie dans le modèle source car il n'est pas directement configuré pour utiliser une source externe pour récupérer la spécification de construction.](#) Généralement, une NO_SOURCE source gère le clonage de tous les référentiels pertinents depuis le buildspec. Pour vous assurer que les informations d'identification configurées sont disponibles pour ces opérations, vous pouvez activer l'`git-credential-helper` option dans le buildspec.

```
env:
  git-credential-helper: yes
```

Au cours de la construction, il CodeBuild lira ensuite le `AuthServer` champ à partir du jeton configuré et utilisera les informations d'identification du jeton pour toutes les requêtes git adressées à ce fournisseur de source tiers en particulier.

Options de configuration supplémentaires

Vous pouvez configurer les informations d'identification au niveau du compte Secrets Manager à l'aide CloudFormation de modèles. Vous pouvez utiliser le CloudFormation modèle suivant pour définir un identifiant au niveau du compte :

```
Parameters:
  GitHubToken:
    Type: String
    NoEcho: true
    Default: placeholder
Resources:
  CodeBuildAuthTokenSecret:
```

```

Type: AWS::SecretsManager::Secret
Properties:
  Description: CodeBuild auth token
  Name: codebuild-auth-token
  SecretString:
    !Join
    - ''
    - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
      - !Ref GitHubToken
      - '"}'
  Tags:
    - Key: codebuild:source:provider
      Value: github
    - Key: codebuild:source:type
      Value: personal_access_token
CodeBuildSecretsManagerAccountCredential:
Type: AWS::CodeBuild::SourceCredential
Properties:
  ServerType: GITHUB
  AuthType: SECRETS_MANAGER
  Token: !Ref CodeBuildAuthTokenSecret

```

Note

Si vous créez également un projet dans la même pile, utilisez l'attribut CloudFormation [DependsOn](#) pour vous assurer qu'AccountCredential est créé avant le projet.

Vous pouvez également configurer plusieurs informations d'identification au niveau de la source de Secrets Manager à l'aide CloudFormation de modèles. Vous pouvez utiliser le CloudFormation modèle suivant pour utiliser plusieurs jetons afin d'extraire plusieurs sources :

```

Parameters:
  GitHubTokenOne:
    Type: String
    NoEcho: true
    Default: placeholder
  GitHubTokenTwo:
    Type: String
    NoEcho: true
    Default: placeholder

```

```
Resources:
  CodeBuildSecretsManagerProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: codebuild-multitoken-example
      ServiceRole: <service-role>
      Environment:
        Type: LINUX_CONTAINER
        ComputeType: BUILD_GENERAL1_SMALL
        Image: aws/codebuild/amazonlinux-x86_64-standard:5.0
      Source:
        Type: GITHUB
        Location: <github-repository-one>
        Auth:
          Type: SECRETS_MANAGER
          Resource: !Ref CodeBuildAuthTokenSecretOne
      SecondarySources:
        - Type: GITHUB
          Location: <github-repository-two>
          Auth:
            Type: SECRETS_MANAGER
            Resource: !Ref CodeBuildAuthTokenSecretTwo
          SourceIdentifier: secondary
      Artifacts:
        Type: NO_ARTIFACTS
      LogsConfig:
        CloudWatchLogs:
          Status: ENABLED
  CodeBuildProjectIAMRoleSecretAccess:
    Type: AWS::IAM::RolePolicy
    Properties:
      RoleName: <role-name>
      PolicyName: CodeBuildProjectIAMRoleSecretAccessPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - secretsmanager:GetSecretValue
            Resource:
              - !Ref CodeBuildAuthTokenSecretOne
              - !Ref CodeBuildAuthTokenSecretTwo
  CodeBuildAuthTokenSecretOne:
    Type: AWS::SecretsManager::Secret
```

```

Properties:
  Description: CodeBuild auth token one
  Name: codebuild-auth-token-one
  SecretString:
    !Join
    - ''
    - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
      - !Ref GitHubTokenOne
      - '"}'
  Tags:
    - Key: codebuild:source:provider
      Value: github
    - Key: codebuild:source:type
      Value: personal_access_token
CodeBuildAuthTokenSecretTwo:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: CodeBuild auth token two
    Name: codebuild-auth-token-two
    SecretString:
      !Join
      - ''
      - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
        - !Ref GitHubTokenTwo
        - '"}'
    Tags:
      - Key: codebuild:source:provider
        Value: github
      - Key: codebuild:source:type
        Value: personal_access_token

```

Supprimer des projets de construction dans AWS CodeBuild

Vous pouvez utiliser la CodeBuild console ou AWS SDKs supprimer un projet de construction dans CodeBuild. AWS CLI Si vous supprimez un projet, ses versions ne sont pas supprimées.

Warning

Vous ne pouvez pas supprimer un projet qui a des générations et une stratégie de ressource. Pour supprimer un projet avec une stratégie de ressources et des générations, supprimez d'abord la stratégie de ressources et ses générations.

Rubriques

- [Suppression d'un projet de génération \(console\)](#)
- [Suppression d'un projet de génération \(AWS CLI\)](#)
- [Suppression d'un projet de génération \(AWS SDKs\)](#)

Suppression d'un projet de génération (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :
 - Sélectionnez la case d'option en regard du projet de génération à supprimer, puis choisissez Supprimer.
 - Sélectionnez le lien pour le projet de génération que vous souhaitez supprimer, puis choisissez Supprimer.

Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour voir plus de projets de génération, choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière pour consulter les projets.

Suppression d'un projet de génération (AWS CLI)

1. Exécutez la commande `delete-project` :

```
aws codebuild delete-project --name name
```

Remplacez l'espace réservé suivant :

- *name*: chaîne obligatoire. Nom du projet de génération à supprimer. Pour obtenir une liste des projets de génération disponibles, exécutez la commande `list-projects`. Pour de plus

amples informations, veuillez consulter [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).

2. En cas de réussite, aucune donnée et aucune erreur n'apparaissent dans la sortie.

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Suppression d'un projet de génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Obtenir un projet de construction public URLs

AWS CodeBuild vous permet de mettre les résultats de construction, les journaux et les artefacts de vos projets de construction à la disposition du grand public. Cela permet aux contributeurs de vos référentiels sources de consulter les résultats et de télécharger les artefacts d'une version, sans avoir besoin d'avoir accès à un AWS compte.

Lorsque vous mettez les versions de votre projet à la disposition du public, tous les résultats, journaux et artefacts d'un projet, y compris les versions exécutées lorsque le projet était privé, sont mis à la disposition du public. De même, lorsque vous rendez privé un projet de build public, les résultats de ce projet ne sont plus accessibles au public.

Pour plus d'informations sur la manière de modifier la visibilité publique des résultats de construction de votre projet, consultez [Permettre l'accès public aux builds](#).

CodeBuild fournit une URL pour les versions publiques de votre projet qui est unique à votre projet.

Pour obtenir l'URL publique de votre projet de build, procédez comme suit.

Pour obtenir l'URL d'un projet de construction public

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Choisissez le lien du projet de construction pour lequel vous souhaitez obtenir l'URL publique.
4. L'URL publique est affichée dans le champ URL publique du projet dans la section Configuration. Vous pouvez choisir le lien pour ouvrir l'URL ou copier l'URL à l'aide du bouton de copie.

Warning

Les points suivants doivent être pris en compte lorsque vous publiez les résultats de construction de votre projet :

- Tous les résultats de construction, les journaux et les artefacts d'un projet, y compris les versions exécutées lorsque le projet était privé, sont accessibles au public.
- Tous les journaux de construction et les artefacts sont accessibles au public. Les variables d'environnement, le code source et d'autres informations sensibles peuvent avoir été générés dans les journaux de construction et les artefacts. Vous devez faire attention aux informations qui sont affichées dans les journaux de construction. Voici certaines des meilleures pratiques :
 - Ne stockez pas de valeurs sensibles, en particulier les clés AWS d'accès IDs et les clés d'accès secrètes, dans les variables d'environnement. Nous vous recommandons d'utiliser un magasin de paramètres Amazon EC2 Systems Manager ou AWS Secrets Manager de stocker des valeurs sensibles.
 - Suivez cette [Bonnes pratiques d'utilisation des webhooks](#) procédure pour limiter les entités qui peuvent déclencher une construction, et ne stockez pas la spécification de construction dans le projet lui-même, afin de garantir que vos webhooks sont aussi sécurisés que possible.
- Un utilisateur malveillant peut utiliser des versions publiques pour distribuer des artefacts malveillants. Nous recommandons aux administrateurs de projet de passer en revue toutes les pull requests afin de vérifier qu'il s'agit d'une modification légitime. Nous vous recommandons également de valider tous les artefacts à l'aide de leurs checksums afin de vous assurer que les bons artefacts sont téléchargés.

Partagez des projets de construction

Le partage de projets permet aux propriétaires de projets de partager leurs AWS CodeBuild projets avec d'autres AWS comptes ou utilisateurs. Dans ce modèle, le compte propriétaire du projet (propriétaire) partage un projet avec d'autres comptes (consommateurs). Un consommateur ne peut pas modifier ou exécuter un projet.

Rubriques

- [Partagez un projet](#)

- [Services connexes](#)
- [Accédez aux CodeBuild projets partagés avec vous](#)
- [Annuler le partage d'un projet partagé](#)
- [Identifier un projet partagé](#)
- [Autorisations de projet partagés](#)

Partagez un projet

Le consommateur peut utiliser à la fois la AWS CodeBuild console AWS CLI et pour visualiser le projet et les versions que vous avez partagés. Le consommateur ne peut pas modifier ou exécuter le projet.

Vous pouvez ajouter un projet à un partage de ressources existant ou en créer un dans la [console AWS RAM](#).

Note

Vous ne pouvez pas supprimer un projet avec des générations qui ont été ajoutées à un partage de ressources.

Pour partager un projet avec des unités organisationnelles ou une organisation entière, vous devez activer le partage avec AWS Organizations. Pour de plus amples informations, veuillez consulter [Activer le partage avec AWS Organizations](#) dans le Guide de l'utilisateur AWS RAM .

Vous pouvez utiliser la AWS CodeBuild console, la AWS RAM console ou le AWS CLI pour partager un projet dont vous êtes le propriétaire.

Prérequis pour le partage de projets

Avant de commencer à partager un projet, assurez-vous que votre AWS compte en est propriétaire. Vous ne pouvez pas partager un projet qui a été partagé avec vous.

Pour partager un projet dont vous êtes propriétaire (CodeBuild console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Dans le volet de navigation, choisissez Projets de génération.

 Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

3. Choisissez le projet que vous souhaitez partager, puis choisissez Partager. Pour de plus amples informations, veuillez consulter [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM .

Pour partager un projet dont vous êtes propriétaire (AWS RAM console)

Consultez [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM .

Pour partager un projet dont vous êtes propriétaire (AWS RAM commande)

Utilisez la commande [create-resource-share](#).

Pour partager un projet que vous possédez (commande CodeBuild)

Utilisez la commande [put-resource-policy](#) :

1. Créez un fichier nommé `policy.json` et copiez ce qui suit dans celui-ci.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource": "arn:aws:iam::*:role/Service*"
  }]
}
```

2. Mettez à jour `policy.json` avec l'ARN du projet et les identificateurs avec qui le partager. L'exemple suivant accorde un accès en lecture seule à l'utilisateur root pour le AWS compte identifié par 123456789012.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    },
    "Action": [
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource": "arn:aws:codebuild:us-west-2:123456789012:project/my-project"
  ]
}
```

3. Exécutez la commande [put-resource-policy](#).

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Obtenez l'ARN du partage de AWS RAM ressources.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Cela renverra une réponse similaire à celle-ci :

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
```

```
    "lastUpdatedTime": "<last-update-time>"
  }
]
}
```

Dans la réponse, copiez la *<resource-share-arn>* valeur à utiliser à l'étape suivante.

5. Exécutez la commande AWS RAM [promote-resource-share-created-from-policy](#).

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-
share-arn>
```

Services connexes

Le partage de projet s'intègre à AWS Resource Access Manager (AWS RAM), un service qui vous permet de partager vos AWS ressources avec n'importe quel AWS compte ou par le biais de n'importe quel compte AWS Organizations. Avec AWS RAM, vous partagez des ressources en créant un partage de ressources qui spécifie les ressources et les consommateurs avec qui les partager. Les consommateurs peuvent être AWS des comptes individuels AWS Organizations, des unités organisationnelles ou une organisation entière AWS Organizations.

Pour plus d'informations, consultez le Guide de l'utilisateur [AWS RAM](#).

Accédez aux CodeBuild projets partagés avec vous

Pour accéder à un projet partagé, le rôle IAM du consommateur nécessite une `BatchGetProjects` autorisation. Vous pouvez associer la politique suivante à leur rôle IAM :

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetProjects"
  ]
}
```

Pour de plus amples informations, veuillez consulter [Utilisation de politiques basées sur l'identité pour AWS CodeBuild](#).

Annuler le partage d'un projet partagé

Un projet non partagé, y compris ses générations, n'est accessible que par son propriétaire. Si vous résiliez le partage d'un projet, les AWS comptes ou utilisateurs avec lesquels vous l'avez précédemment partagé ne peuvent pas accéder au projet ou à ses versions.

Pour annuler le partage d'un projet partagé qui vous appartient, vous devez le supprimer du partage de ressources. Vous pouvez utiliser la AWS CodeBuild console, AWS RAM la console ou AWS CLI pour ce faire.

Pour annuler le partage d'un projet partagé dont vous êtes le propriétaire (AWS RAM console)

Consultez la section [Mise à jour d'un partage de ressources](#) du Guide de l'utilisateur AWS RAM .

Pour annuler le partage d'un projet partagé que vous possédez (AWS CLI)

Utilisez la commande [disassociate-resource-share](#).

Pour annuler le partage d'un projet dont vous êtes propriétaire (CodeBuild commande)

Exécutez la [delete-resource-policy](#) commande et spécifiez l'ARN du projet dont vous souhaitez annuler le partage :

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Identifier un projet partagé

Les propriétaires et les consommateurs peuvent les utiliser AWS CLI pour identifier les projets partagés.

Pour identifier les projets partagés avec votre AWS compte ou utilisateur (AWS CLI)

Utilisez la [list-shared-projects](#) commande pour renvoyer les projets partagés avec vous.

Autorisations de projet partagés

Autorisations accordées aux propriétaires

Un propriétaire de projet peut modifier le projet et l'utiliser pour exécuter des générations.

Autorisations accordées aux consommateurs

Un consommateur de projet peut afficher un projet et ses versions, mais ne peut pas modifier un projet ou l'utiliser pour exécuter des générations.

Tag : projets de construction

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou AWS assignez à une AWS ressource. Chaque AWS étiquette comporte deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment`, `Project` ou `Secret`). Les touches de tag distinguent les majuscules et minuscules.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333`, `Production` ou le nom d'une équipe). Omettre la valeur de balise équivaut à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Ensemble, ces éléments sont connus sous le nom de paires clé-valeur. Pour plus d'informations sur le nombre de balises que vous pouvez avoir sur un projet et les restrictions sur les clés et les valeurs de balises, veuillez consulter [Étiquettes](#).

Les balises vous aident à identifier et à organiser vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à un CodeBuild projet que celle que vous attribuez à un compartiment S3. Pour plus d'informations sur l'utilisation des balises, consultez la section [Bonnes pratiques en matière de balisage](#).

Dans CodeBuild, les ressources principales sont le projet et le groupe de rapports. Vous pouvez utiliser la CodeBuild console, le AWS CLI CodeBuild APIs, ou AWS SDKs pour ajouter, gérer et supprimer des balises pour un projet. Outre l'identification, l'organisation et le suivi de votre projet à l'aide de balises, vous pouvez utiliser des balises dans les politiques IAM pour contrôler les personnes autorisées à consulter votre projet et à interagir avec celui-ci. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Important

Lorsque vous utilisez la fonctionnalité de capacité réservée, les données mises en cache sur les instances de flotte, y compris les fichiers source, les couches Docker et les répertoires mis en cache spécifiés dans les spécifications de construction, peuvent être accessibles aux autres projets du même compte. Cela est intentionnel et permet aux projets au sein d'un même compte de partager des instances de flotte.

Rubriques

- [Ajouter une balise à un projet](#)
- [Afficher les balises d'un projet](#)
- [Modifier les balises d'un projet](#)
- [Supprimer une balise d'un projet](#)

Ajouter une balise à un projet

L'ajout de balises à un projet peut vous aider à identifier et à organiser vos AWS ressources et à gérer l'accès à celles-ci. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à un projet. N'oubliez pas qu'il y a des limites sur le nombre de balises que vous pouvez avoir dans un projet. Il existe des restrictions sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour de plus amples informations, veuillez consulter [Étiquettes](#). Une fois que vous avez des balises, vous pouvez créer des politiques IAM pour gérer l'accès au projet en fonction de ces balises. Vous pouvez utiliser la CodeBuild console ou le AWS CLI pour ajouter des balises à un projet.

Important

Lorsque vous utilisez la fonctionnalité de capacité réservée, les données mises en cache sur les instances de flotte, y compris les fichiers source, les couches Docker et les répertoires mis en cache spécifiés dans les spécifications de construction, peuvent être accessibles aux autres projets du même compte. Cela est intentionnel et permet aux projets au sein d'un même compte de partager des instances de flotte.

Pour plus d'informations sur l'ajout de balises à un projet lorsque vous le créez, consultez [Ajouter une balise à un projet \(console\)](#).

Important

Avant d'ajouter une balise à un projet, assurez-vous de passer en revue toutes les politiques IAM susceptibles d'utiliser des balises pour contrôler l'accès aux ressources, telles que les projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Rubriques

- [Ajouter une balise à un projet \(console\)](#)
- [Ajouter une balise à un projet \(AWS CLI\)](#)

Ajouter une balise à un projet (console)

Vous pouvez utiliser la CodeBuild console pour ajouter une ou plusieurs balises à un CodeBuild projet.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Build projects (Créer des projets), choisissez le nom du projet auquel vous souhaitez ajouter des balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Si aucune balise n'a été ajoutée au projet, choisissez Ajouter une balise. Sinon, choisissez Edit (Modifier), puis choisissez Add tag (Ajouter une balise).
5. Dans Key (Clé), entrez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).
6. (Facultatif) Pour ajouter une autre balise, choisissez à nouveau Add tag (Ajouter une balise).
7. Lorsque vous avez fini d'ajouter des balises, choisissez Submit (Envoyer).

Ajouter une balise à un projet (AWS CLI)

Pour ajouter une balise à un pipeline lors de sa création, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#). Dans `create-project.json`, ajoutez vos balises.

Dans ces étapes, nous supposons que vous avez déjà installé une version récente de l' AWS CLI ou que vous avez procédé à une mise à jour vers la version actuelle. Pour plus d'informations, consultez [Installing the AWS Command Line Interface](#) (Installation de).

Si elle aboutit, cette commande ne renvoie rien.

Afficher les balises d'un projet

Les balises peuvent vous aider à identifier et à organiser vos AWS ressources, ainsi qu'à gérer l'accès à celles-ci. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc sur les [bonnes pratiques de balisage](#). Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Afficher les balises d'un projet (console)

Vous pouvez utiliser la CodeBuild console pour afficher les balises associées à un CodeBuild projet.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Build projects (Créer des projets), choisissez le nom du projet dans lequel vous souhaitez afficher les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).

Afficher les balises d'un projet (AWS CLI)

Pour afficher les balises d'un projet de génération, exécutez la commande suivante. Utilisez le nom de votre projet pour le paramètre `--names`.

```
aws codebuild batch-get-projects --names your-project-name
```

Si elle réussit, cette commande renvoie des informations au format JSON sur votre projet de construction qui incluent quelque chose comme ce qui suit :

```
{
  "tags": {
    "Status": "Secret",
    "Team": "JanesProject"
```

```
}  
}
```

Si le projet n'a pas de balises, la section `tags` est vide :

```
"tags": []
```

Modifier les balises d'un projet

Vous pouvez modifier la valeur d'une balise associée à un projet. Vous pouvez également modifier le nom de la clé, ce qui équivaut à supprimer la balise et à ajoutant une carte différente avec le nouveau nom et la même valeur que l'autre clé. N'oubliez pas qu'il y a des limites sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour de plus amples informations, veuillez consulter [Étiquettes](#).

Important

La modification des balises d'un projet peut avoir un impact sur l'accès au projet. Avant de modifier le nom (clé) ou la valeur d'une balise pour un projet, assurez-vous de passer en revue les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources telles que les projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Modifier une balise d'un projet (console)

Vous pouvez utiliser la CodeBuild console pour modifier les balises associées à un CodeBuild projet.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Build projects (Créer des projets), choisissez le nom du projet dont vous souhaitez modifier les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Choisissez Modifier.
5. Effectuez l'une des actions suivantes :

- Pour modifier la balise, entrez un nouveau nom dans Key (Clé). La modification du nom de la balise est l'équivalent de la suppression d'une balise et de l'ajout d'une nouvelle balise avec le nouveau nom de clé.
 - Pour modifier la valeur d'une balise, saisissez une nouvelle valeur. Si vous souhaitez modifier la valeur en valeur nulle, supprimez la valeur actuelle et laissez le champ vide.
6. Lorsque vous avez terminé de modifier des balises, choisissez Submit (Soumettre).

Modifier les balises d'un projet (AWS CLI)

Pour ajouter, modifier ou supprimer des balises d'un projet de génération, veuillez consulter [Modification des paramètres d'un projet de génération \(AWS CLI\)](#). Mettez à jour la section tags des données au format JSON que vous utilisez pour mettre à jour le projet.

Supprimer une balise d'un projet

Vous pouvez supprimer une ou plusieurs balises associées à un projet. La suppression d'un tag ne supprime pas le tag des autres AWS ressources associées à ce tag.

Important

La suppression des balises d'un projet peut avoir un impact sur l'accès au projet. Avant de supprimer une balise d'un projet, assurez-vous de passer en revue toutes les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources telles que les projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Supprimer une balise d'un projet (console)

Vous pouvez utiliser la CodeBuild console pour supprimer l'association entre un tag et un CodeBuild projet.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Build projects (Créer des projets), choisissez le nom du projet dans lequel vous souhaitez supprimer les balises.

3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Choisissez Modifier.
5. Trouvez la balise que vous voulez supprimer, puis choisissez Remove tag (Supprimer une balise).
6. Lorsque vous avez terminé de supprimer les balises, choisissez Submit (Envoyer).

Supprimer une balise d'un projet (AWS CLI)

Pour supprimer une ou plusieurs balises d'un projet de génération, reportez-vous à la section [Modification des paramètres d'un projet de génération \(AWS CLI\)](#). Mettez à jour la tags section dans les données formatées JSON avec une liste mise à jour de balises qui ne contient pas celles que vous souhaitez supprimer. Si vous souhaitez supprimer toutes les balises, mettez à jour la section tags pour :

```
"tags: []"
```

Note

Si vous supprimez un projet de CodeBuild génération, toutes les associations de balises sont supprimées du projet de génération supprimé. Vous n'avez pas besoin de supprimer les balises avant de supprimer un projet de génération.

Utilisez des coureurs avec AWS CodeBuild

AWS CodeBuild prend en charge l'intégration avec les coureurs GitHub Actions, les GitLab coureurs autogérés et le Buildkite Runner.

Rubriques

- [GitHub Actions auto-hébergées se lance dans AWS CodeBuild](#)
- [GitLab Coureurs autogérés en AWS CodeBuild](#)
- [Buildkite Runner autogéré dans AWS CodeBuild](#)

GitHub Actions auto-hébergées se lance dans AWS CodeBuild

Vous pouvez configurer votre projet pour configurer des exécuteurs d' GitHub actions auto-hébergés dans des CodeBuild conteneurs afin de traiter vos tâches de flux de travail GitHub Actions. Cela peut être fait en configurant un webhook à l'aide de votre CodeBuild projet et en mettant à jour votre flux de travail GitHub Actions YAML pour utiliser des coureurs auto-hébergés hébergés sur des machines. CodeBuild

Les étapes de haut niveau pour configurer un CodeBuild projet afin d'exécuter GitHub des tâches Actions sont les suivantes :

1. Si ce n'est pas déjà fait, créez un jeton d'accès personnel ou connectez-vous à une OAuth application à laquelle connecter votre projet GitHub.
2. Accédez à la CodeBuild console, créez un CodeBuild projet avec un webhook et configurez vos filtres webhook.
3. Mettez à jour votre flux de travail GitHub Actions YAML GitHub pour configurer votre environnement de génération.

Pour une procédure plus détaillée, voir [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

Cette fonctionnalité permet à vos tâches de flux de travail GitHub Actions de s'intégrer de manière native AWS, ce qui apporte sécurité et commodité grâce à des fonctionnalités telles que l'IAM AWS CloudTrail, AWS Secrets Manager l'intégration et Amazon VPC. Vous pouvez accéder aux derniers types d'instances, y compris les instances basées sur ARM.

Rubriques

- [À propos de l'application GitHub Actions CodeBuild Runner hébergée](#)
- [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#)
- [Résoudre les problèmes liés au webhook](#)
- [Les remplacements d'étiquettes sont pris en charge avec le programme Actions Runner CodeBuild hébergé par -hosted GitHub](#)
- [Calculez les images prises en charge avec le CodeBuild lanceur d' GitHub actions hébergé](#)

À propos de l'application GitHub Actions CodeBuild Runner hébergée

Voici quelques questions fréquemment posées à propos du lanceur d' GitHub actions CodeBuild hébergé par -hosted.

Quand dois-je inclure les remplacements d'image et d'instance dans l'étiquette ?

Vous pouvez inclure les remplacements d'image et d'instance dans l'étiquette afin de spécifier un environnement de génération différent pour chacune de vos tâches de flux de travail GitHub Actions. Cela peut être fait sans qu'il soit nécessaire de créer plusieurs CodeBuild projets ou webhooks. Par exemple, cela est utile lorsque vous devez utiliser une [matrice pour vos tâches de flux de travail](#).

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      image:${{ matrix.os }}
      instance-size:${{ matrix.size }}
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

Note

Les guillemets peuvent être nécessaires si `runs-on` plusieurs libellés contiennent le contexte GitHub des actions.

Puis-je utiliser CloudFormation cette fonctionnalité ?

Oui, vous pouvez inclure un groupe de filtres dans votre CloudFormation modèle qui spécifie un filtre d'événements de travail du flux de travail GitHub Actions dans le webhook de votre projet.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Pour de plus amples informations, veuillez consulter [Filtrer les événements du GitHub webhook \(CloudFormation\)](#).

Si vous avez besoin d'aide pour configurer les informations d'identification du projet dans votre CloudFormation modèle, consultez [AWS::CodeBuild::SourceCredential](#) le guide de AWS CloudFormation l'utilisateur pour plus d'informations.

Comment masquer des secrets lors de l'utilisation de cette fonctionnalité ?

Par défaut, les secrets imprimés dans le journal ne sont pas masqués. Si vous souhaitez masquer vos secrets, vous pouvez utiliser la syntaxe suivante : `::add-mask::` *value*. Voici un exemple de la façon dont vous pouvez utiliser cette syntaxe dans votre fichier YAML :

```
name: Secret Job
on: [push]
jobs:
  Secret-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    env:
      SECRET_NAME: "secret-name"
    steps:
      - run: echo "::add-mask::$SECRET_NAME"
```

Pour plus d'informations, consultez la section [Masquage d'une valeur lors d'une connexion](#). GitHub

Puis-je recevoir des événements Webhook GitHub Actions provenant de plusieurs référentiels au sein d'un même projet ?

CodeBuild prend en charge les webhooks au niveau des organisations et au niveau mondial, qui reçoivent des événements d'une organisation ou d'une entreprise spécifiée. Pour de plus amples informations, veuillez consulter [GitHub webhooks mondiaux et organisationnels](#).

Quelles régions proposent l'utilisation d'un CodeBuild programme GitHub Actions Runner hébergé ?

CodeBuild-Hosted GitHub Actions : les coureurs sont soutenus dans toutes les CodeBuild régions. Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Quelles plateformes prennent en charge l'utilisation d'un CodeBuild exécuteur d' GitHub actions hébergé ?

CodeBuild-hosted GitHub Actions Runners sont pris en charge à la fois sur Amazon EC2 [AWS Lambda](#) et Compute. Vous pouvez utiliser les plateformes suivantes : Amazon Linux 2, Amazon Linux 2023, Ubuntu et Windows Server Core 2019. Pour plus d'informations, consultez [EC2 calculer des images](#) et [Images de calcul Lambda](#).

Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé

Ce didacticiel explique comment configurer vos CodeBuild projets pour exécuter GitHub des tâches Actions. Pour plus d'informations sur l'utilisation d' GitHub Actions avec, CodeBuild voir [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

Pour effectuer ce didacticiel, vous devez d'abord :

- Connectez-vous à l'aide d'un jeton d'accès personnel, d'un secret du Gestionnaire de Secrets, d'une OAuth application ou d'une GitHub appli. Si vous souhaitez vous connecter à une OAuth application, vous devez utiliser la CodeBuild console pour ce faire. Si vous souhaitez créer un jeton d'accès personnel, vous pouvez utiliser la CodeBuild console ou l'[ImportSourceCredentials API](#). Pour plus d'informations, consultez [GitHub et accès au serveur d' GitHub entreprise dans CodeBuild](#).
- Connectez-vous CodeBuild à votre GitHub compte. Pour ce faire, vous pouvez effectuer l'une des opérations suivantes :
 - Vous pouvez l'ajouter GitHub en tant que fournisseur de source dans la console. Vous pouvez vous connecter à l'aide d'un jeton d'accès personnel, d'un secret du Gestionnaire de Secrets, d'une OAuth application ou d'une GitHub application. Pour obtenir des instructions, veuillez consulter [GitHub et accès au serveur d' GitHub entreprise dans CodeBuild](#).
 - Vous pouvez importer vos GitHub informations d'identification via l'[ImportSourceCredentials API](#). Cela ne peut être fait qu'avec un jeton d'accès personnel. Si vous vous connectez à l'aide d'une OAuth application, vous devez plutôt vous connecter à l'aide de la console. Pour obtenir des instructions, veuillez consulter [Connectez-vous à GitHub l'aide d'un jeton d'accès \(CLI\)](#).

 Note

Cela ne doit être fait que si vous n'êtes pas connecté GitHub à votre compte.

Étape 1 : Création d'un CodeBuild projet avec un webhook

Au cours de cette étape, vous allez créer un CodeBuild projet avec un webhook et le passer en revue dans la GitHub console. Vous pouvez également choisir GitHub Enterprise comme fournisseur source. Pour en savoir plus sur la création d'un webhook dans GitHub Enterprise, consultez [GitHub webhooks manuels](#).

Pour créer un CodeBuild projet avec un webhook

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
3. Dans Type de projet, choisissez le projet Runner.

Dans Runner :

- a. Pour le fournisseur Runner, choisissez GitHub.
- b. Pour l'emplacement de Runner, choisissez Repository.
- c. Pour URL du référentiel sous Repository, choisissez <https://github.com/user-name/repository-name>.

 Note

Par défaut, votre projet ne recevra que les `WORKFLOW_JOB_QUEUED` événements d'un seul référentiel. Si vous souhaitez recevoir des événements pour tous les référentiels d'une organisation ou d'une entreprise, consultez [GitHub webhooks mondiaux et organisationnels](#).

4. • Dans Environment (Environnement) :

- Choisissez une image d'environnement compatible et calculez. Notez que vous avez la possibilité de remplacer les paramètres d'image et d'instance en utilisant une étiquette dans le code YAML de votre flux de travail GitHub Actions. Pour de plus amples informations, consultez [Étape 2 : mettez à jour votre flux de travail GitHub Actions YAML](#).
 - Dans Buildspec:
 - Notez que votre spécification de construction sera ignorée à moins qu'elle ne `buildspec-override:true` soit ajoutée sous forme d'étiquette. Au lieu de cela, il le CodeBuild remplacera pour utiliser des commandes qui configureront le coureur auto-hébergé.
5. Continuez avec les valeurs par défaut, puis choisissez Create build project.
 6. Ouvrez la GitHub console sur <https://github.com/user-name/repository-name/settings/hooks> pour vérifier qu'un webhook a été créé et qu'il est activé pour diffuser des événements de tâches Workflow.

Étape 2 : mettez à jour votre flux de travail GitHub Actions YAML

Au cours de cette étape, vous allez mettre à jour le fichier YAML de votre flux de travail GitHub Actions [GitHub](#) pour configurer votre environnement de construction et utiliser les coureurs auto-hébergés par GitHub Actions dans. CodeBuild Pour plus d'informations, consultez les sections [Utilisation d'étiquettes avec des coureurs auto-hébergés](#) et [Les remplacements d'étiquettes sont pris en charge avec le programme Actions Runner CodeBuild hébergé par -hosted GitHub](#).

Mettez à jour votre flux de travail GitHub Actions (YAML)

Accédez au [runs-on](#) paramètre YAML de votre flux de travail GitHub Actions [GitHub](#) et mettez-le à jour pour configurer votre environnement de génération. Pour ce faire, vous pouvez effectuer l'une des opérations suivantes :

- Vous pouvez spécifier le nom du projet et l'ID d'exécution, auquel cas le build utilisera la configuration de votre projet existante pour le calcul, l'image, la version de l'image et la taille de l'instance. Le nom du projet est nécessaire pour lier AWS les paramètres associés de votre tâche GitHub Actions à un CodeBuild projet spécifique. En incluant le nom du projet dans le YAML, CodeBuild il est autorisé à invoquer des tâches avec les paramètres de projet corrects. En fournissant l'ID d'exécution, votre build CodeBuild sera mappé à des exécutions de flux de travail spécifiques et arrêtera la génération lorsque l'exécution du flux de travail est annulée. Pour plus d'informations, reportez-vous à la section [githubContexte](#).

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

 Note

Assurez-vous que vous *<project-name>* correspondez au nom du projet que vous avez créé à l'étape précédente. S'il ne correspond pas, le webhook de CodeBuild sera pas traité et le flux de travail GitHub Actions risque de se bloquer.

Voici un exemple de flux de travail GitHub Actions YAML :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    steps:
      - run: echo "Hello World!"
```

- Vous pouvez également remplacer votre image et le type de calcul dans l'étiquette. Consultez [Calculez les images prises en charge avec le CodeBuild lanceur d' GitHub actions hébergé](#) la liste des images sélectionnées. Pour utiliser des images personnalisées, voir [Les remplacements d'étiquettes sont pris en charge avec le programme Actions Runner CodeBuild hébergé par - hosted GitHub](#). Le type de calcul et l'image figurant dans l'étiquette remplaceront les paramètres d'environnement de votre projet. Pour remplacer les paramètres de votre environnement pour une version de calcul CodeBuild EC2 ou Lambda, utilisez la syntaxe suivante :

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
  image:<environment-type>-<image-identifiant>
  instance-size:<instance-size>
```

Voici un exemple de flux de travail GitHub Actions YAML :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
```

```
image:arm-3.0
instance-size:small
steps:
- run: echo "Hello World!"
```

- Vous pouvez remplacer le parc utilisé pour votre construction dans l'étiquette. Cela remplacera les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour de plus amples informations, veuillez consulter [Exécutez des builds sur des flottes à capacité réservée](#). Pour remplacer les paramètres de votre flotte pour une version de calcul Amazon EC2, utilisez la syntaxe suivante :

```
runs-on:
- codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
fleet:<fleet-name>
```

Pour remplacer à la fois la flotte et l'image utilisées pour la génération, utilisez la syntaxe suivante :

```
runs-on:
- codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
fleet:<fleet-name>
image:<environment-type>-<image-identifiant>
```

Voici un exemple de flux de travail GitHub Actions YAML :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      fleet:myFleet
      image:arm-3.0
    steps:
      - run: echo "Hello World!"
```

- Pour exécuter vos tâches GitHub Actions sur une image personnalisée, vous pouvez configurer une image personnalisée dans votre CodeBuild projet et éviter de fournir une étiquette de remplacement d'image. CodeBuild utilisera l'image configurée dans le projet si aucune étiquette de remplacement d'image n'est fournie.

- Vous pouvez éventuellement fournir des étiquettes autres que celles prises CodeBuild en charge. Ces étiquettes seront ignorées dans le but de remplacer les attributs de la version, mais elles n'échoueront pas à la demande de webhook. Par exemple, l'ajout `testLabel` d'une étiquette n'empêchera pas le build de s'exécuter.

Note

Si une dépendance fournie par GitHub -hosted runners n'est pas disponible dans l'CodeBuild environnement, vous pouvez l'installer à l'aide d' GitHub Actions dans votre flux de travail. Par exemple, vous pouvez utiliser l'[setup-python](#) action pour installer Python dans votre environnement de construction.

Exécutez les commandes buildspec pendant les phases `INSTALL`, `PRE_BUILD` et `POST_BUILD`

Par défaut, CodeBuild ignore les commandes buildspec lors de l'exécution d'une version Actions auto-hébergée. GitHub Pour exécuter les commandes buildspec pendant la construction, `buildspec-override:true` vous pouvez les ajouter en tant que suffixe à l'étiquette :

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    buildspec-override:true
```

En utilisant cette commande, CodeBuild vous créerez un dossier appelé `actions-runner` dans le dossier source principal du conteneur. Lorsque le lanceur d' GitHub actions démarre pendant la `BUILD` phase, il s'exécute dans le `actions-runner` répertoire.

L'utilisation d'une dérogation buildspec dans une version Actions auto-hébergée présente plusieurs limites : GitHub

- CodeBuild n'exécutera pas de commandes buildspec pendant la `BUILD` phase, car le lanceur auto-hébergé s'exécute pendant la phase. `BUILD`
- CodeBuild ne téléchargera aucune source principale ou secondaire pendant la `DOWNLOAD_SOURCE` phase. Si vous avez configuré un fichier buildspec, seul ce fichier sera téléchargé depuis la source principale du projet.
- Si une commande de génération échoue pendant la `INSTALL` phase `PRE_BUILD` ou, elle ne CodeBuild démarrera pas le lanceur auto-hébergé et le travail du flux de travail GitHub Actions devra être annulé manuellement.

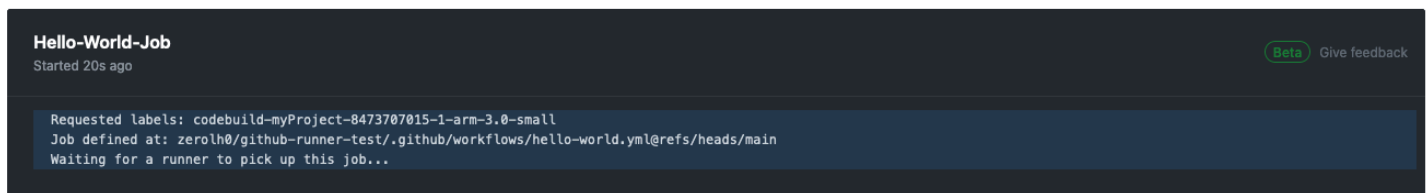
- CodeBuild récupère le jeton du coureur pendant la `DOWNLOAD_SOURCE` phase, dont le délai d'expiration est d'une heure. Si votre `PRE_BUILD` ou vos `INSTALL` phases dépassent une heure, le jeton de course peut expirer avant le départ du coureur GitHub auto-hébergé.

Étape 3 : Passez en revue vos résultats

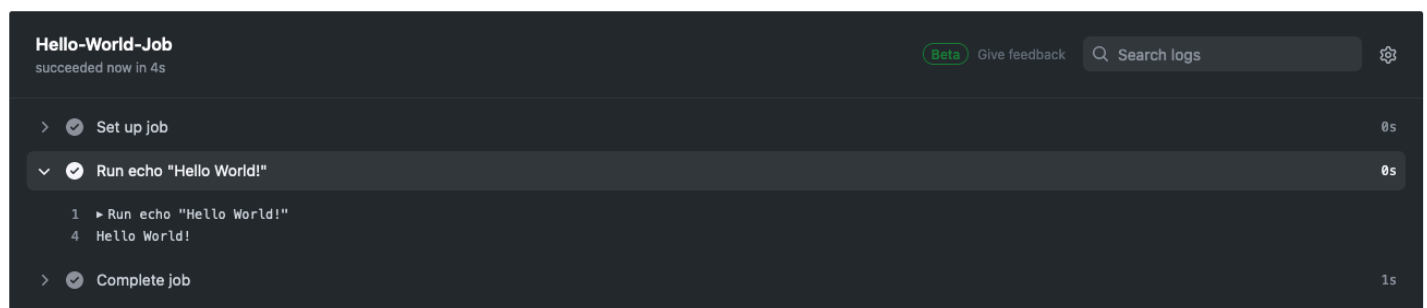
Chaque fois qu'un flux de travail GitHub Actions est exécuté, CodeBuild il reçoit les événements de travail du flux de travail via le webhook. Pour chaque tâche du flux de travail, CodeBuild lance une compilation pour exécuter un exécuteur d' GitHub actions éphémère. Le coureur est responsable de l'exécution d'une seule tâche de flux de travail. Une fois le travail terminé, le lanceur et le processus de construction associé seront immédiatement interrompus.

Pour consulter les journaux des tâches de votre flux de travail GitHub, accédez à votre référentiel dans, choisissez Actions, choisissez le flux de travail souhaité, puis choisissez le travail spécifique dont vous souhaitez consulter les journaux.

Vous pouvez consulter les étiquettes demandées dans le journal pendant que le travail attend d'être récupéré par un coureur auto-hébergé. CodeBuild



Une fois la tâche terminée, vous pourrez consulter le journal de la tâche.



GitHub Options de configuration d'Actions Runner

Vous pouvez spécifier les variables d'environnement suivantes dans la configuration de votre projet pour modifier la configuration de vos coureurs auto-hébergés.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

CodeBuild enregistrera les coureurs auto-hébergés au nom de l'organisation spécifié comme valeur de cette variable d'environnement. Pour plus d'informations sur l'enregistrement des coureurs au niveau de l'organisation et sur les autorisations nécessaires, voir [Créer une configuration pour un just-in-time coureur au sein d'une organisation](#).

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

CodeBuild enregistrera les coureurs auto-hébergés sous le nom d'entreprise spécifié comme valeur de cette variable d'environnement. Pour plus d'informations sur l'enregistrement des coureurs au niveau de l'entreprise et sur les autorisations nécessaires, voir [Créer une configuration pour un just-in-time coureur pour une entreprise](#).

Note

Par défaut, les Enterprise Runners ne sont pas disponibles dans les référentiels de l'organisation. Pour que les coureurs auto-hébergés puissent prendre en charge les tâches liées au flux de travail, vous devrez peut-être configurer les paramètres d'accès de votre groupe de coureurs. Pour plus d'informations, consultez la section Mise [à disposition des Enterprise Runners dans les référentiels](#).

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

CodeBuild enregistrera les coureurs auto-hébergés selon l'identifiant entier du groupe de coureurs stocké sous forme de valeur de cette variable d'environnement. Par défaut, cette valeur est 1. Pour plus d'informations sur les groupes de coureurs auto-hébergés, voir [Gestion de l'accès aux coureurs auto-hébergés à l'aide](#) de groupes.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

Pour configurer l'enregistrement des coureurs au niveau de l'organisation à l'aide du fichier YAML de votre flux de travail GitHub Actions, vous pouvez utiliser la syntaxe suivante :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
```



```
- codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
  organization-registration-name:myOrganization
steps:
- run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

Pour configurer l'enregistrement des coureurs au niveau de l'entreprise à l'aide du fichier YAML de votre flux de travail GitHub Actions, vous pouvez utiliser la syntaxe suivante :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        enterprise-registration-name:myEnterprise
    steps:
      - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

Pour configurer l'enregistrement des coureurs à un identifiant de groupe de coureurs spécifique à l'aide du fichier YAML de votre flux de travail GitHub Actions, vous pouvez utiliser la syntaxe suivante :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        registration-group-id:3
    steps:
      - run: echo "Hello World!"
```

Filtrer GitHub les actions et les événements du webhook ()CloudFormation

La partie suivante d'un CloudFormation modèle au format YAML crée un groupe de filtres qui déclenche une génération lorsqu'elle est évaluée à true. Le groupe de filtres suivant spécifie une

demande de travail de flux de travail GitHub Actions avec un nom de flux de travail correspondant à l'expression régulière `\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
    FilterGroups:
      - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
      - Type: WORKFLOW_NAME
        Pattern: \[CI-CodeBuild\]
```

Filtrer GitHub les actions et les événements du webhook ()AWS CDK

Le AWS CDK modèle suivant crée un groupe de filtres qui déclenche une génération lorsqu'il est évalué comme vrai. Le groupe de filtres suivant spécifie une demande de travail dans le flux de travail GitHub Actions.

```
import { aws_codebuild as codebuild } from 'aws-cdk-lib';
import { EventAction, FilterGroup } from "aws-cdk-lib/aws-codebuild";

const source = codebuild.Source.gitHub({
  owner: 'owner',
  repo: 'repo',
  webhook: true,
  webhookFilters: [FilterGroup.inEventOf(EventAction.WORKFLOW_JOB_QUEUED)],
```

```
})
```

Filtrer GitHub les actions et les événements du webhook (Terraform)

Le modèle Terraform suivant crée un groupe de filtres qui déclenche une génération lorsqu'il est évalué comme vrai. Le groupe de filtres suivant spécifie une demande de travail dans le flux de travail GitHub Actions.

```
resource "aws_codebuild_webhook" "example" {
  project_name = aws_codebuild_project.example.name
  build_type   = "BUILD"
  filter_group {
    filter {
      type      = "EVENT"
      pattern   = "WORKFLOW_JOB_QUEUED"
    }
  }
}
```

Filtrer GitHub les actions et les événements du webhook (AWS CLI)

Les AWS CLI commandes suivantes créent un projet GitHub Actions Runner auto-hébergé avec un groupe de filtres de demandes de travail GitHub Actions Workflow qui déclenche une compilation lorsqu'elle est évaluée comme vraie.

```
aws codebuild create-project \
--name <project name> \
--source "{\"type\":\"GITHUB\",\"location\":\"<repository location>\",\"buildspec\":\"\\\"\\\"}\" \
--artifacts "{\"type\":\"NO_ARTIFACTS\"}" \
--environment "{\"type\":\"LINUX_CONTAINER\",\"image\":\"aws/codebuild/amazonlinux-x86_64-standard:5.0\",\"computeType\":\"BUILD_GENERAL1_MEDIUM\"}" \
--service-role "<service role ARN>"
```

```
aws codebuild create-webhook \
--project-name <project name> \
--filter-groups "[{\"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]"
```

Résoudre les problèmes liés au webhook

Problème : le webhook que vous avez configuré ne fonctionne [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#) pas ou votre tâche de flux de travail est en GitHub suspens.

Causes possibles :

- Votre événement Webhook Workflow jobs ne parvient peut-être pas à déclencher une compilation. Consultez les journaux des réponses pour voir la réponse ou le message d'erreur.
- Vos tâches sont attribuées au mauvais agent d'expédition en raison de la configuration de son étiquette. Ce problème peut se produire lorsque l'une de vos tâches au cours d'une seule exécution de flux de travail comporte moins d'étiquettes qu'une autre tâche. Par exemple, si vous avez deux tâches portant les étiquettes suivantes dans le même flux de travail, exécutez :
 - Job 1 : `codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`
 - Job 2 : `codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`,
`instance-size:medium`

Lors du routage d'une tâche GitHub Actions auto-hébergée, la tâche GitHub sera acheminée vers n'importe quel exécuteur avec toutes les étiquettes spécifiées pour la tâche. Ce comportement signifie que le Job 1 peut être choisi par le coureur créé pour le Job 1 ou Job 2, mais que le Job 2 ne peut être repris que par le coureur créé pour le Job 2 puisqu'il possède une étiquette supplémentaire. Si le Job 1 est choisi par le coureur créé pour le Job 2, le Job 2 sera bloqué car le coureur du Job 1 n'a pas le `instance-size:medium` label.

Solutions recommandées :

Lorsque vous créez plusieurs tâches dans le même flux de travail, utilisez le même nombre de remplacements d'étiquettes pour chaque tâche ou attribuez à chaque tâche une étiquette personnalisée, telle que `job1` ou `job2`.

Si l'erreur persiste, suivez les instructions ci-dessous pour résoudre le problème.

1. Ouvrez la GitHub console à l'adresse <https://github.com/user-name/repository-name/settings/hooks> pour afficher les paramètres du webhook de votre dépôt. Sur cette page, vous verrez un webhook créé pour votre dépôt.
2. Choisissez Modifier et confirmez que le webhook est activé pour diffuser les événements relatifs aux jobs Workflow.

☐ **Team adds**
Team added or modified on a repository.

☐ **Visibility changes**
Repository changes from private to public.

☐ **Watches**
User stars a repository.

☐ **Wiki**
Wiki page updated.

☒ **Workflow jobs**
Workflow job queued, waiting, in progress, or completed on a repository.

☐ **Workflow runs**
Workflow run requested or completed on a repository.

☒ **Active**
We will deliver event details when this hook is triggered.

Update webhook

Delete webhook

- Accédez à l'onglet Livraisons récentes, recherchez l'`workflow_job.queued` événement correspondant et développez l'événement.
- Vérifiez le champ des étiquettes dans la charge utile et assurez-vous qu'il correspond à vos attentes.
- Enfin, passez en revue l'onglet Réponse, car il contient la réponse ou le message d'erreur renvoyé par CodeBuild.

Settings

Recent Deliveries

⚠️

workflow_job.queued

2024-10-01 14:20:21

Request

Response **400**

Redeliver

⌚ Completed in 0.00 seconds.

Headers

- Vous pouvez également déboguer les échecs de webhook à l'aide GitHub de. APIs Vous pouvez consulter les livraisons récentes d'un webhook à l'aide de l'API [Lister les livraisons pour un webhook de référentiel](#) :

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-API-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries
```

Après avoir trouvé la livraison du webhook que vous souhaitez déboguer et avoir noté l'ID de livraison, vous pouvez utiliser l'API [Get a delivery for a repository webhook](#). CodeBuild la réponse à la charge utile de livraison du webhook se trouve dans la `response` section :

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-API-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries/delivery-id
```

Problème : vos GitHub actions lorsque les règles de [protection du déploiement](#) sont activées déclenchent des déclencheurs CodeBuild intégrés avant que le déploiement ne soit approuvé.

Causes possibles : CodeBuild récupère le déploiement et l'environnement associés à la tâche GitHub Actions s'ils existent pour vérifier s'ils sont approuvés. Si vous CodeBuild ne parvenez pas à récupérer le déploiement ou l'environnement, la CodeBuild compilation peut être déclenchée prématurément.

Solutions recommandées : Vérifiez que les informations d'identification associées à vos CodeBuild projets disposent d'autorisations de lecture pour les déploiements et les actions dans le cadre GitHub de ces projets.

Les remplacements d'étiquettes sont pris en charge avec le programme Actions Runner CodeBuild hébergé par -hosted GitHub

Dans votre flux de travail GitHub Actions YAML, vous pouvez fournir diverses remplacements d'étiquettes qui modifient la version de votre runner auto-hébergée. Toutes les versions non reconnues par CodeBuild seront ignorées mais n'échoueront pas à votre demande de webhook. Par exemple, le flux de travail YAML suivant inclut des remplacements pour l'image, la taille de l'instance, le parc et les spécifications de construction :

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:${{ matrix.os }}
        instance-size:${{ matrix.size }}
```

```
fleet:myFleet
buildspec-override:true
strategy:
  matrix:
    include:
      - os: arm-3.0
        size: small
      - os: linux-5.0
        size: large
  steps:
    - run: echo "Hello World!"
```

Note

Si votre tâche de flux de travail est en suspens GitHub, reportez-vous [Résoudre les problèmes liés au webhook](#) à la section [Utilisation d'étiquettes personnalisées pour acheminer les tâches](#).

codebuild-*<project-name>*-\${{github.run_id}}-\${{github.run_attempt}}

(obligatoire)

- Exemple : codebuild-fake-project-\${{ github.run_id }}-\${{ github.run_attempt }}
- Obligatoire pour tous les flux de travail GitHub Actions YAMLS. *<project name>* doit être égal au nom du projet pour lequel le runner webhook auto-hébergé est configuré.

image:*<environment-type>*-*<image-identifiant>*

- Exemple : image:arm-3.0
- Remplace l'image et le type d'environnement utilisés lors du démarrage de la version auto-hébergée par Runner avec une image organisée. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculez les images prises en charge avec le CodeBuild lanceur d' GitHub actions hébergé](#).
 - Pour remplacer l'image et le type d'environnement utilisés par une image personnalisée, utilisez image:custom-*<environment-type>*-*<custom-image-identifiant>*
 - Exemple : image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0

 Note

Si l'image personnalisée se trouve dans un registre privé, consultez [Configurer un identifiant de registre privé pour les coureurs auto-hébergés](#).

`instance-size:` *<instance-size>*

- Exemple : `instance-size:medium`
- Remplace le type d'instance utilisé lors du démarrage de la version de course auto-hébergée. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculez les images prises en charge avec le CodeBuild lanceur d' GitHub actions hébergé](#).

`fleet:` *<fleet-name>*

- Exemple : `fleet:myFleet`
- Remplace les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour de plus amples informations, veuillez consulter [Exécutez des builds sur des flottes à capacité réservée](#).

`buildspec-override:` *<boolean>*

- Exemple : `buildspec-override:true`
- Permet à la compilation d'exécuter des commandes buildspec dans les POST_BUILD phases INSTALLPRE_BUILD, et si elle est définie sur. `true`

Déroger d'une seule étiquette (ancienne version)

CodeBuild vous permet de fournir plusieurs remplacements dans une seule étiquette en utilisant ce qui suit :

- Pour remplacer les paramètres de votre environnement pour une version de calcul Amazon EC2/Lambda, utilisez la syntaxe suivante :

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-  
${{ github.run_attempt }}-<environment-type>-<image-identifiant>-<instance-size>
```


- Pour modifier les paramètres de votre flotte pour Amazon EC2 Compute Build, utilisez la syntaxe suivante :

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}-
fleet-<fleet-name>
```

- Pour remplacer à la fois la flotte et l'image utilisées pour la génération, utilisez la syntaxe suivante :

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-image-<image-version>-fleet-<fleet-name>
```

- Pour exécuter les commandes builds spec pendant la construction, `-with-buildspec` vous pouvez les ajouter en tant que suffixe à l'étiquette :

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>-with-buildspec
```

- Vous pouvez éventuellement modifier la taille de l'instance sans remplacer l'image. Pour les versions Amazon EC2, vous pouvez exclure à la fois le type d'environnement et l'identifiant d'image. Pour les versions Lambda, vous pouvez exclure l'identifiant de l'image.

Calculez les images prises en charge avec le CodeBuild lanceur d' GitHub actions hébergé

Dans l'étiquette que vous avez configurée [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#), vous pouvez remplacer les paramètres de votre environnement Amazon EC2 en utilisant les valeurs des trois premières colonnes. CodeBuild fournit les images de calcul Amazon EC2 suivantes. Pour plus d'informations sur

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-sta	al/standa rd/4.0
		medium			
		large			

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
		xlarge		ndard:4.0	
linux	5.0	2xlarge gpu_small gpu_large	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
linux-ec2	latest	small medium large	Amazon Linux 2	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Aucune
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2,0
arm	3.0	2xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3,0

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
arm-ec2	latest	small medium large	Amazon Linux 2	aws/codebuild/ami/amazonlinux-arm-base:latest	Aucune
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	Ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	Ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	Ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/A
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Aucune

En outre, vous pouvez modifier les paramètres de votre environnement Lambda en utilisant les valeurs suivantes. Pour plus d'informations sur le calcul CodeBuild Lambda, consultez [Exécuter des builds sur AWS Lambda ordinateur](#). CodeBuild prend en charge les images de calcul Lambda suivantes :

Type d'environnement	Identificateur de l'image	Taille d'instance			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto11	4GB			
		8GB			
	corretto17	10GB			
	corretto21				
	nodejs18				

Type d'environnement	Identificateur de l'image	Taille d'instance			
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Pour plus d'informations, consultez [Modes et types de calcul de l'environnement de création](#) et [Images Docker fournies par CodeBuild](#).

GitLab Coureurs autogérés en AWS CodeBuild

GitLab propose deux modes d'exécution pour exécuter GitLab des tâches dans votre CI/CD pipeline. L'un des modes est celui des coureurs GitLab hébergés, qui sont gérés par GitLab et entièrement intégrés à GitLab. L'autre mode est celui des coureurs autogérés, qui vous permet de créer votre propre environnement personnalisé pour exécuter des tâches dans le pipeline GitLab CI/CD.

Les étapes de haut niveau pour configurer un CodeBuild projet afin d'exécuter des tâches de pipeline GitLab CI/CD sont les suivantes :

1. Si ce n'est pas déjà fait, connectez-vous à l'aide d'une OAuth application à laquelle connecter votre projet GitLab.
2. Accédez à la CodeBuild console, créez un CodeBuild projet avec un webhook et configurez vos filtres webhook.
3. Mettez à jour votre pipeline GitLab CI/CD YAML GitLab pour configurer votre environnement de construction.

Pour une procédure plus détaillée, voir [Tutoriel : Configuration d'un CodeBuild coureur hébergé GitLab](#).

Cette fonctionnalité permet à vos tâches de pipeline GitLab CI/CD de s'intégrer de manière native AWS, ce qui garantit sécurité et commodité grâce à des fonctionnalités telles que IAM AWS CloudTrail et Amazon VPC. Vous pouvez accéder aux derniers types d'instances, y compris les instances basées sur ARM.

Rubriques

- [À propos du CodeBuild coureur hébergé GitLab](#)
- [Tutoriel : Configuration d'un CodeBuild coureur hébergé GitLab](#)
- [Les remplacements d'étiquettes sont pris en charge par le CodeBuild coureur hébergé GitLab](#)
- [Calculer les images prises en charge par le logiciel CodeBuild -hosted runner GitLab](#)

À propos du CodeBuild coureur hébergé GitLab

Voici quelques questions courantes concernant le CodeBuild GitLab coureur hébergé.

Quels types de sources sont pris en charge pour les GitLab coureurs CodeBuild hébergés ?

CodeBuild GitLab -les coureurs hébergés sont pris en charge pour le type GITLAB et GITLAB_SELF_MANAGED source.

Quand dois-je inclure les remplacements d'image et d'instance dans l'étiquette ?

Vous pouvez inclure les remplacements d'image et d'instance dans l'étiquette afin de spécifier un environnement de construction différent pour chacune de vos tâches de pipeline GitLab CI/CD. Cela peut être fait sans qu'il soit nécessaire de créer plusieurs CodeBuild projets ou webhooks.

Puis-je utiliser CloudFormation cette fonctionnalité ?

Oui, vous pouvez inclure un groupe de filtres dans votre CloudFormation modèle qui spécifie un GitLab filtre d'événements de travail dans le webhook de votre projet.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Pour de plus amples informations, veuillez consulter [Filtrer les événements du GitLab webhook \(CloudFormation\)](#).

Si vous avez besoin d'aide pour configurer les informations d'identification du projet dans votre CloudFormation modèle, consultez [AWS::CodeBuild::SourceCredential](#) le guide de AWS CloudFormation l'utilisateur pour plus d'informations.

Comment masquer des secrets lors de l'utilisation de cette fonctionnalité ?

Par défaut, les secrets imprimés dans le journal ne sont pas masqués. Si vous souhaitez masquer vos secrets, vous pouvez le faire en mettant à jour les paramètres de vos variables d' CI/CD environnement :

Pour masquer des secrets GitLab

1. Dans vos GitLab paramètres, choisissez CI/CD.
2. Dans Variables, choisissez Modifier pour le secret que vous souhaitez masquer.
3. Dans Visibilité, sélectionnez Variable de masque, puis choisissez Mettre à jour la variable pour enregistrer vos modifications.

Puis-je recevoir des événements GitLab webhook issus de plusieurs projets au sein d'un même groupe ?

CodeBuild prend en charge les webhooks de groupe, qui reçoivent des événements d'un GitLab groupe spécifié. Pour de plus amples informations, veuillez consulter [GitLab webhooks de groupe](#).

Puis-je exécuter une tâche dans Docker Executor pour le runner autogéré ? Par exemple, je souhaite exécuter une tâche de pipeline sur une image spécifique afin de conserver le même environnement de construction dans un conteneur séparé et isolé.

Vous pouvez exécuter le GitLab runner autogéré CodeBuild avec une image spécifique en [créant le projet avec une image personnalisée](#) ou en [remplaçant l'image](#) dans votre `.gitlab-ci.yml` fichier.

Avec quel exécuteur testamentaire CodeBuild court le coureur autogéré ?

Le runner autogéré CodeBuild s'exécute avec l'exécuteur shell, où le build s'exécute localement avec le GitLab runner qui s'exécute dans le conteneur docker.

Puis-je fournir des commandes buildspec avec le runner autogéré ?

Oui, il est possible d'ajouter des commandes buildspec avec un runner autogéré. Vous pouvez fournir le fichier buildspec.yml dans votre GitLab dépôt et utiliser la `buildspec-override:true` balise dans la section Tags de la tâche. Pour de plus amples informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).

Quelles sont les régions où l'utilisation d'un GitLab coureur CodeBuild hébergé est prise en charge ?

CodeBuild GitLab -les coureurs hébergés sont soutenus dans toutes les CodeBuild régions. Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Quelles plateformes prennent en charge l'utilisation d'un CodeBuild GitLab coureur hébergé ?

CodeBuild GitLab -les coureurs hébergés sont pris en charge à la fois sur Amazon EC2 [AWS Lambda](#) et sur Compute. Vous pouvez utiliser les plateformes suivantes : Amazon Linux 2, Amazon Linux 2023, Ubuntu et Windows Server Core 2019. Pour plus d'informations, consultez [EC2 calculer des images](#) et [Images de calcul Lambda](#).

Tutoriel : Configuration d'un CodeBuild coureur hébergé GitLab

Ce didacticiel explique comment configurer vos CodeBuild projets pour exécuter des tâches de pipeline GitLab CI/CD. Pour plus d'informations sur l'utilisation GitLab ou l' GitLab autogestion avec CodeBuild, consultez [GitLab Coureurs autogérés en AWS CodeBuild](#).

Pour effectuer ce didacticiel, vous devez d'abord :

- Connectez-vous à une OAuth application en utilisant CodeConnections. Notez que lorsque vous vous connectez à une OAuth application, vous devez utiliser la CodeBuild console pour ce faire. Pour plus d'informations, consultez [GitLab accéder à CodeBuild](#).
- Connectez-vous CodeBuild à votre GitLab compte. Pour ce faire, vous pouvez l'ajouter GitLab en tant que fournisseur de source dans la console. Pour obtenir des instructions, veuillez consulter [GitLab accéder à CodeBuild](#).

Note

Cela ne doit être fait que si vous n'êtes pas connecté GitLab à votre compte. Cette fonctionnalité CodeBuild nécessite des autorisations supplémentaires, telles que `create_runner` et `manage_runner` depuis l' GitLab OAuth application. S'il en existe CodeConnections pour un GitLab compte en particulier, celui-ci ne demande pas automatiquement de mises à jour des autorisations. Pour ce faire, vous pouvez accéder à la CodeConnections console et créer une connexion fictive avec le même GitLab compte pour déclencher la réautorisation afin d'obtenir les autorisations supplémentaires. Ainsi, toutes les connexions existantes peuvent utiliser la fonction Runner. Une fois l'opération terminée, vous pouvez supprimer la connexion fictive.

Étape 1 : Création d'un CodeBuild projet avec un webhook

Au cours de cette étape, vous allez créer un CodeBuild projet avec un webhook et le passer en revue dans la GitLab console.

Pour créer un CodeBuild projet avec un webhook

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).

Dans Type de projet, choisissez le projet Runner.

- Dans Runner :
 - Pour le fournisseur Runner, choisissez GitLab.
 - Pour Credential, sélectionnez l'une des options suivantes :
 - Choisissez les informations d'identification de source par défaut. La connexion par défaut applique une GitLab connexion par défaut à tous les projets.
 - Choisissez Informations d'identification source personnalisées. La connexion personnalisée applique une GitLab connexion personnalisée qui remplace les paramètres par défaut de votre compte.

Note

Si vous n'avez pas encore créé de connexion avec votre fournisseur, vous devrez créer une nouvelle GitLab connexion. Pour obtenir des instructions, veuillez consulter [Connect CodeBuild à GitLab](#).

- Pour l'emplacement de Runner, choisissez Repository.
- Pour Repository, choisissez le nom de votre projet en GitLab spécifiant le chemin du projet avec l'espace de noms.
- Dans Environment (Environnement) :
 - Choisissez une image d'environnement compatible et calculez. Notez que vous avez la possibilité de remplacer les paramètres d'image et d'instance en utilisant une étiquette dans le YAML de votre pipeline GitLab CI/CD. Pour de plus amples informations, veuillez consulter [Étape 2 : Créez un fichier .gitlab-ci.yml dans votre dépôt](#).

- Dans Buildspec:
 - Notez que votre spécification de construction sera ignorée à moins qu'elle ne `buildspec-override:true` soit ajoutée sous forme d'étiquette. Au lieu de cela, il le CodeBuild remplacera pour utiliser des commandes qui configureront le coureur autogéré.
 -
3. Continuez avec les valeurs par défaut, puis choisissez Create build project.
 4. Ouvrez la GitLab console sur <https://gitlab.com/user-name/repository-name/-/hooks> pour vérifier qu'un webhook a été créé et qu'il est activé pour diffuser des événements de tâches Workflow.

Étape 2 : Créez un fichier .gitlab-ci.yml dans votre dépôt

Au cours de cette étape, vous allez créer un `.gitlab-ci.yml` fichier dans lequel vous [GitLabalisez](#) configurer votre environnement de construction et utiliser des runners GitLab autogérés. CodeBuild Pour plus d'informations, consultez la section [Utiliser des coureurs autogérés](#).

Mettez à jour votre pipeline GitLab CI/CD YAML

Accédez à votre référentiel <https://gitlab.com/user-name/project-name/-/tree/branch-name> et créez-en un `.gitlab-ci.yml`. Vous pouvez configurer votre environnement de génération en effectuant l'une des opérations suivantes :

- Vous pouvez spécifier le nom du CodeBuild projet, auquel cas le build utilisera votre configuration de projet existante pour le calcul, l'image, la version de l'image et la taille de l'instance. Le nom du projet est nécessaire pour lier les AWS paramètres associés à votre GitLab tâche à un CodeBuild projet spécifique. En incluant le nom du projet dans le YAML, CodeBuild il est autorisé à invoquer des tâches avec les paramètres de projet corrects.

```
tags:  
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

`$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME` est nécessaire pour mapper la construction à des exécutions de tâches de pipeline spécifiques et arrêter la génération lorsque l'exécution du pipeline est annulée.

Note

Assurez-vous que le nom du projet que vous avez créé *<project-name>* correspond à celui dans lequel vous l'avez créé CodeBuild. S'il ne correspond pas, le webhook ne CodeBuild sera pas traité et le pipeline GitLab CI/CD risque de se bloquer.

Voici un exemple de pipeline GitLab CI/CD YAML :

```
workflow:
  name: HelloWorld
stages:          # List of stages for jobs, and their order of execution
  - build

build-job:       # This job runs in the build stage, which runs first.
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

- Vous pouvez également remplacer votre image et le type de calcul dans la balise. Consultez [Calculer les images prises en charge par le logiciel CodeBuild -hosted runner GitLab](#) la liste des images sélectionnées. Pour utiliser des images personnalisées, voir [Les remplacements d'étiquettes sont pris en charge par le CodeBuild coureur hébergé GitLab](#). Le type de calcul et l'image contenus dans la balise remplaceront les paramètres d'environnement de votre projet. Pour remplacer les paramètres de votre environnement pour une version de calcul Amazon EC2, utilisez la syntaxe suivante :

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - image:<environment-type>-<image-identifiant>
  - instance-size:<instance-size>
```

Voici un exemple de pipeline GitLab CI/CD YAML :

```
stages:
  - build
```

```
build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small
```

- Vous pouvez remplacer la flotte utilisée pour votre build dans le tag. Cela remplacera les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour de plus amples informations, veuillez consulter [Exécutez des builds sur des flottes à capacité réservée](#). Pour remplacer les paramètres de votre flotte pour une version de calcul Amazon EC2, utilisez la syntaxe suivante :

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>
```

Pour remplacer à la fois la flotte et l'image utilisées pour la génération, utilisez la syntaxe suivante :

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>
  - image:<environment-type>-<image-identifiant>
```

Voici un exemple de pipeline GitLab CI/CD YAML :

```
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:myFleet
    - image:arm-3.0
```

- Pour exécuter vos tâches de pipeline GitLab CI/CD sur une image personnalisée, vous pouvez configurer une image personnalisée dans votre CodeBuild projet et éviter de fournir une étiquette de remplacement d'image. CodeBuild utilisera l'image configurée dans le projet si aucune étiquette de remplacement d'image n'est fournie.

Une fois que vous aurez validé vos modifications `.gitlab-ci.yml`, un GitLab pipeline sera déclenché et une notification Webhook sera envoyée pour démarrer votre build in CodeBuild.
`build-job`

Exécutez les commandes `buildspec` pendant les phases `INSTALL`, `PRE_BUILD` et `POST_BUILD`

Par défaut, CodeBuild ignore les commandes `buildspec` lors de l'exécution d'une version autogérée. GitLab Pour exécuter les commandes `buildspec` pendant la construction, `buildspec-override:true` vous pouvez les ajouter en tant que suffixe à : `tags`

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - buildspec-override:true
```

En utilisant cette commande, CodeBuild vous créerez un dossier appelé `gitlab-runner` dans le dossier source principal du conteneur. Lorsque le GitLab coureur démarre pendant la `BUILD` phase, il court dans le `gitlab-runner` répertoire.

L'utilisation d'une dérogation `buildspec` dans une version autogérée présente plusieurs limites :

GitLab

- CodeBuild n'exécutera pas de commandes `buildspec` pendant la `BUILD` phase, car le lanceur autogéré s'exécute pendant la phase. `BUILD`
- CodeBuild ne téléchargera aucune source principale ou secondaire pendant la `DOWNLOAD_SOURCE` phase. Si vous avez configuré un fichier `buildspec`, seul ce fichier sera téléchargé depuis la source principale du projet.
- Si une commande de construction échoue pendant la `INSTALL` phase `PRE_BUILD` ou, elle ne CodeBuild démarrera pas le lanceur autogéré et la tâche du pipeline GitLab CI/CD devra être annulée manuellement.
- CodeBuild récupère le jeton du coureur pendant la `DOWNLOAD_SOURCE` phase, dont le délai d'expiration est d'une heure. Si votre `PRE_BUILD` ou vos `INSTALL` phases dépassent une heure, le jeton de course peut expirer avant le départ du coureur GitLab autogéré.

Étape 3 : Passez en revue vos résultats

Chaque fois qu'un GitLab CI/CD pipeline run occurs, CodeBuild would receive the CI/CD pipeline job events through the webhook. For each job in the CI/CD pipeline, CodeBuild starts a build to run an ephemeral GitLab runner. The runner is responsible for executing a single CI/CD projet de pipeline est effectué. Une fois le travail terminé, le lanceur et le processus de construction associé seront immédiatement interrompus.

Pour consulter les journaux des tâches de votre CI/CD pipeline, accédez à votre référentiel GitLab, choisissez Build, Jobs, puis choisissez le Job spécifique dont vous souhaitez consulter les journaux.

Vous pouvez consulter les étiquettes demandées dans le journal pendant que le travail attend d'être récupéré par un coureur autogéré. CodeBuild

Filtrer les événements du GitLab webhook ()CloudFormation

La partie suivante d'un CloudFormation modèle au format YAML crée un groupe de filtres qui déclenche une génération lorsqu'elle est évaluée à true. Le groupe de filtres suivant spécifie un nom de GitLab CI/CD pipeline job request with a CI/CD pipeline correspondant à l'expression régulière\[CI-CodeBuild\].

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
      FilterGroups:
        - - Type: EVENT
```

```
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]
```

Les remplacements d'étiquettes sont pris en charge par le CodeBuild coureur hébergé GitLab

Dans votre pipeline GitLab CI/CD YAML, vous pouvez fournir une variété de remplacements d'étiquettes qui modifient votre build de runner autogéré. Toutes les versions non reconnues par CodeBuild seront ignorées mais n'échoueront pas à votre demande de webhook. Par exemple, le code YAML suivant inclut les remplacements relatifs à l'image, à la taille de l'instance, au parc et aux spécifications de construction :

```
workflow:
  name: HelloWorld
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small
    - fleet:myFleet
    - buildspec-override:true
```

codebuild-*<project-name>*-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME
(obligatoire)

- Exemple : codebuild-myProject-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME
- Nécessaire pour tous les GitLab pipelines CI/CD. YAMLS *<project name>* doit être égal au nom du projet pour lequel le runner webhook autogéré est configuré.

image:*<environment-type>*-*<image-identifiant>*

- Exemple : image:arm-3.0

- Remplace l'image et le type d'environnement utilisés lors du démarrage de la version autogérée du runner. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculer les images prises en charge par le logiciel CodeBuild -hosted runner GitLab](#).
- Pour remplacer l'image et le type d'environnement utilisés par une image personnalisée, utilisez `image:custom-<environment-type>-<custom-image-identifiant>`
- Exemple : `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

 Note

Si l'image personnalisée se trouve dans un registre privé, consultez [Configurer un identifiant de registre privé pour les coureurs auto-hébergés](#).

`instance-size:<instance-size>`

- Exemple : `instance-size:small`
- Remplace le type d'instance utilisé lors du démarrage de la version autogérée du runner. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculer les images prises en charge par le logiciel CodeBuild -hosted runner GitLab](#).

`fleet:<fleet-name>`

- Exemple : `fleet:myFleet`
- Remplace les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour de plus amples informations, veuillez consulter [Exécutez des builds sur des flottes à capacité réservée](#).

`buildspec-override:<boolean>`

- Exemple : `buildspec-override:true`
- Permet à la compilation d'exécuter des commandes buildspec dans les POST_BUILD phases INSTALLPRE_BUILD, et si elle est définie sur `true`

Calculer les images prises en charge par le logiciel CodeBuild -hosted runner GitLab

Dans l'étiquette que vous avez configurée [Tutoriel : Configuration d'un CodeBuild coureur hébergé GitLab](#), vous pouvez remplacer les paramètres de votre environnement Amazon EC2 en utilisant les valeurs des trois premières colonnes. CodeBuild fournit les images de calcul Amazon EC2 suivantes. Pour plus d'informations sur

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image	Définition
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
		gpu_small			
		gpu_large			
linux-ec2	latest	small	Amazon Linux 2	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Aucune
		medium			
		large			
arm	2.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-armv7-standard:2.0	al/aarch64/standard/2.0
		medium			
		large			
		xlarge			

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image	Définition
		2xlarge		standard:2.0	
arm	3.0		Amazon Linux 2	aws/codebuild/amazonlinux-arm64-standard:3.0	al/aarch64/standard/3,0
arm-ec2	latest	small medium large	Amazon Linux 2	aws/codebuild/ami/amazonlinux-arm-base:latest	Aucune
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	Ubuntu/standard/5.0
ubuntu	6.0	large xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	Ubuntu/standard/6.0
ubuntu	7.0	2xlarge gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	Ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image	Définition
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/A
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Aucune

En outre, vous pouvez modifier les paramètres de votre environnement Lambda en utilisant les valeurs suivantes. Pour plus d'informations sur le calcul CodeBuild Lambda, consultez [Exécuter des builds sur AWS Lambda ordinateur](#). CodeBuild prend en charge les images de calcul Lambda suivantes :

Type d'environnement	Version d'environnement d'exécution	Taille d'instance			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			

Type d'environnement	Version d'environnement d'exécution	Taille d'instance			
arm-lambda	corretto11	4GB			
		8GB			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Pour plus d'informations, consultez [Modes et types de calcul de l'environnement de création](#) et [Images Docker fournies par CodeBuild](#).

Buildkite Runner autogéré dans AWS CodeBuild

Vous pouvez configurer votre projet pour configurer des moteurs Buildkite auto-hébergés dans des CodeBuild conteneurs afin de traiter vos tâches Buildkite. Cela peut être fait en configurant un webhook à l'aide de votre CodeBuild projet et en mettant à jour les étapes YAML de votre pipeline Buildkite pour utiliser des coureurs auto-hébergés hébergés sur des machines. CodeBuild

Les étapes de haut niveau pour configurer un CodeBuild projet afin d'exécuter des tâches Buildkite sont les suivantes :

- Accédez à la CodeBuild console et créez un CodeBuild projet avec la configuration de type Buildkite Runner project runner
- Ajoutez un `job.scheduled` webhook à votre organisation Buildkite.
- Mettez à jour les étapes YAML de votre pipeline Buildkite dans Buildkite pour configurer votre environnement de construction.

Pour une procédure plus détaillée, voir [Tutoriel : Configuration d'un CodeBuild moteur Buildkite hébergé](#). Cette fonctionnalité permet à vos tâches Buildkite de s'intégrer de manière native AWS, ce qui garantit sécurité et commodité grâce à des fonctionnalités telles que IAM, AWS Secrets Manager, AWS CloudTrail et Amazon VPC. Vous pouvez accéder aux derniers types d'instances, y compris les instances basées sur ARM.

À propos du CodeBuild Buildkite Runner hébergé

Voici quelques questions courantes concernant le Buildkite CodeBuild Runner hébergé.

Quand dois-je inclure les remplacements d'image et d'instance dans l'étiquette ?

Vous pouvez inclure les remplacements d'image et d'instance dans l'étiquette afin de spécifier un environnement de construction différent pour chacune de vos tâches Buildkite. Cela peut être fait sans qu'il soit nécessaire de créer plusieurs CodeBuild projets ou webhooks. Par exemple, cela est utile lorsque vous devez utiliser une [matrice pour les tâches Buildkite](#).

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "${matrix.os}"
      instance-size: "${matrix.size}"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "a12-5.0"
        size:
          - "small"
          - "large"
```

Est-il possible de CodeBuild créer automatiquement des webhooks dans Buildkite ?

Actuellement, Buildkite exige que tous les webhooks soient créés manuellement à l'aide de leur console. Vous pouvez suivre le didacticiel sur [Tutoriel : Configuration d'un CodeBuild moteur Buildkite hébergé](#) pour créer un webhook Buildkite manuellement dans la console Buildkite.

Puis-je les utiliser CloudFormation pour créer des webhooks Buildkite ?

CloudFormation n'est actuellement pas pris en charge pour les webhooks Buildkite Runner, car Buildkite nécessite que les webhooks soient créés manuellement à l'aide de leur console.

Quelles régions sont compatibles avec l'utilisation d'un CodeBuild Buildkite Runner hébergé ?

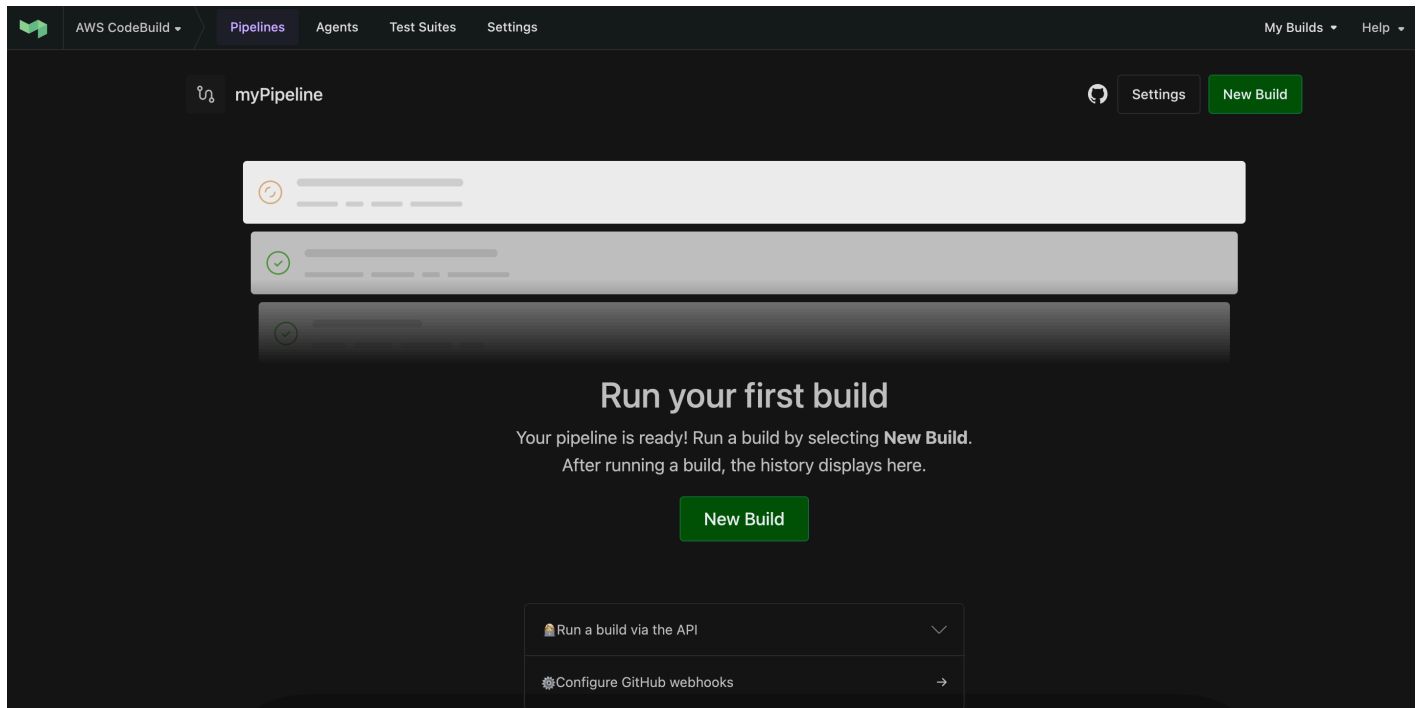
CodeBuild- les coureurs Buildkite hébergés sont pris en charge dans toutes les régions. CodeBuild Pour plus d'informations sur AWS les régions où cette CodeBuild option est disponible, consultez la section [AWS Services par région](#).

Tutoriel : Configuration d'un CodeBuild moteur Buildkite hébergé

Ce didacticiel vous montre comment configurer vos CodeBuild projets pour exécuter des tâches Buildkite. Pour plus d'informations sur l'utilisation de Buildkite avec CodeBuild , voir. [Buildkite Runner autogéré dans AWS CodeBuild](#)

Pour effectuer ce didacticiel, vous devez d'abord :

- Ayez accès à une organisation Buildkite. Pour plus d'informations sur la configuration d'un compte et d'une organisation Buildkite, vous pouvez suivre ce didacticiel de [démarrage](#).
- Créez un pipeline, un cluster et une file d'attente Buildkite configurés pour utiliser des coureurs auto-hébergés. Pour plus d'informations sur la configuration de ces ressources, vous pouvez consulter le didacticiel de [configuration du pipeline Buildkite](#).



Étape 1 : générer un jeton d'agent Buildkite

Au cours de cette étape, vous allez générer un jeton d'agent dans Buildkite qui sera utilisé pour authentifier les coureurs auto-hébergés. CodeBuild Pour plus d'informations sur cette ressource, consultez [Buildkite Agent Tokens](#).

Pour générer un jeton d'agent Buildkite

1. Dans votre cluster Buildkite, choisissez Agent Tokens, puis choisissez New Token.
2. Ajoutez une description au jeton et cliquez sur Créer un jeton.
3. Enregistrez la valeur du jeton de l'agent, car elle sera utilisée ultérieurement lors de la configuration CodeBuild du projet.

Agent Tokens > New

Description — Required

myToken

Describe the set of agents this token is for

Allowed IP Addresses

0.0.0.0/0 ::/0

Restrict which network addresses are allowed to use this agent token. Use space-separated, IPv4 **CIDR notation**, for example:

192.0.2.0/24 198.51.100.12 25c7:c056:9943:1e01::/64 .

Create Token

Étape 2 : Création d'un CodeBuild projet avec un webhook

Pour créer un CodeBuild projet avec un webhook

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de construction auto-hébergé. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Configuration du projet, sélectionnez le projet Runner. Dans Runner :
 - Pour le fournisseur Runner, choisissez Buildkite.
 - Pour le jeton d'agent Buildkite, choisissez Créer un nouveau jeton d'agent en utilisant la page de création d'un secret. Vous serez invité à créer un nouveau secret AWS Secrets Manager avec une valeur de secret égale au jeton d'agent Buildkite que vous avez généré ci-dessus.
 - (Facultatif) Si vous souhaitez utiliser des informations d'identification CodeBuild gérées pour votre tâche, sélectionnez le fournisseur du référentiel source de votre tâche dans les options d'identification source de Buildkite et vérifiez que les informations d'identification sont configurées pour votre compte. Vérifiez également que votre pipeline Buildkite utilise Checkout via HTTPS.

 Note

Buildkite nécessite des informations d'identification de source dans l'environnement de construction afin d'extraire le code source de votre tâche. Consultez [Authentification de Buildkite auprès d'un dépôt privé](#) les options d'identification de source disponibles.

- (Facultatif) Dans l'environnement :
 - Choisissez une image d'environnement compatible et calculez.

Notez que vous avez la possibilité de remplacer les paramètres d'image et d'instance en utilisant une étiquette dans les étapes YAML de Buildkite. Pour de plus amples informations, veuillez consulter [Étape 4 : Mettez à jour les étapes de votre pipeline Buildkite](#).

- (Facultatif) Dans Buildspec :
 - Votre spécification de construction sera ignorée par défaut à moins qu'elle ne `buildspec-override: "true"` soit ajoutée en tant qu'étiquette. Au lieu de cela, il le CodeBuild remplacera pour utiliser des commandes qui configureront le coureur auto-hébergé.

 Note

CodeBuild ne prend pas en charge les fichiers buildspec pour les versions de Runner auto-hébergées par Buildkite. Pour les spécifications de construction en ligne, vous devrez les activer [git-credential-helper](#) dans votre spécification de construction si vous avez configuré les informations d'identification des sources gérées CodeBuild

3. Continuez avec les valeurs par défaut, puis choisissez Create build project.
4. Enregistrez l'URL de charge utile et les valeurs secrètes dans la fenêtre contextuelle Create Webhook. Suivez les instructions affichées dans la fenêtre contextuelle pour créer un nouveau webhook d'organisation Buildkite ou passez à la section suivante.

Étape 3 : créer un CodeBuild webhook dans Buildkite

Au cours de cette étape, vous utiliserez l'URL de charge utile et les valeurs secrètes du CodeBuild webhook pour créer un nouveau webhook dans Buildkite. Ce webhook sera utilisé pour déclencher des builds au CodeBuild moment où une tâche Buildkite valide démarre.

Pour créer un nouveau webhook dans Buildkite

1. Accédez à la page des paramètres de votre organisation Buildkite.
2. Sous Intégrations, sélectionnez Services de notification.
3. Choisissez Ajouter à côté de la zone Webhook. Sur la page Ajouter une notification Webhook, utilisez la configuration suivante :
 - a. Sous URL du webhook, ajoutez la valeur de l'URL de charge utile enregistrée.
 - b. Sous Jeton, vérifiez que l'option Envoyer le jeton sous X-Buildkite-Token est sélectionnée. Ajoutez la valeur secrète de votre webhook dans le champ Token.
 - c. Sous, vérifiez que l'option Envoyer le jeton sous X-Buildkite-Token est sélectionnée. Ajoutez la valeur secrète de votre webhook dans le champ Token.
 - d. Sous Events, sélectionnez l'événement `job.scheduled` webhook.
 - e. (Facultatif) Sous Pipelines, vous pouvez éventuellement choisir de ne déclencher des builds que pour un pipeline spécifique.
4. Choisissez Ajouter une notification Webhook.

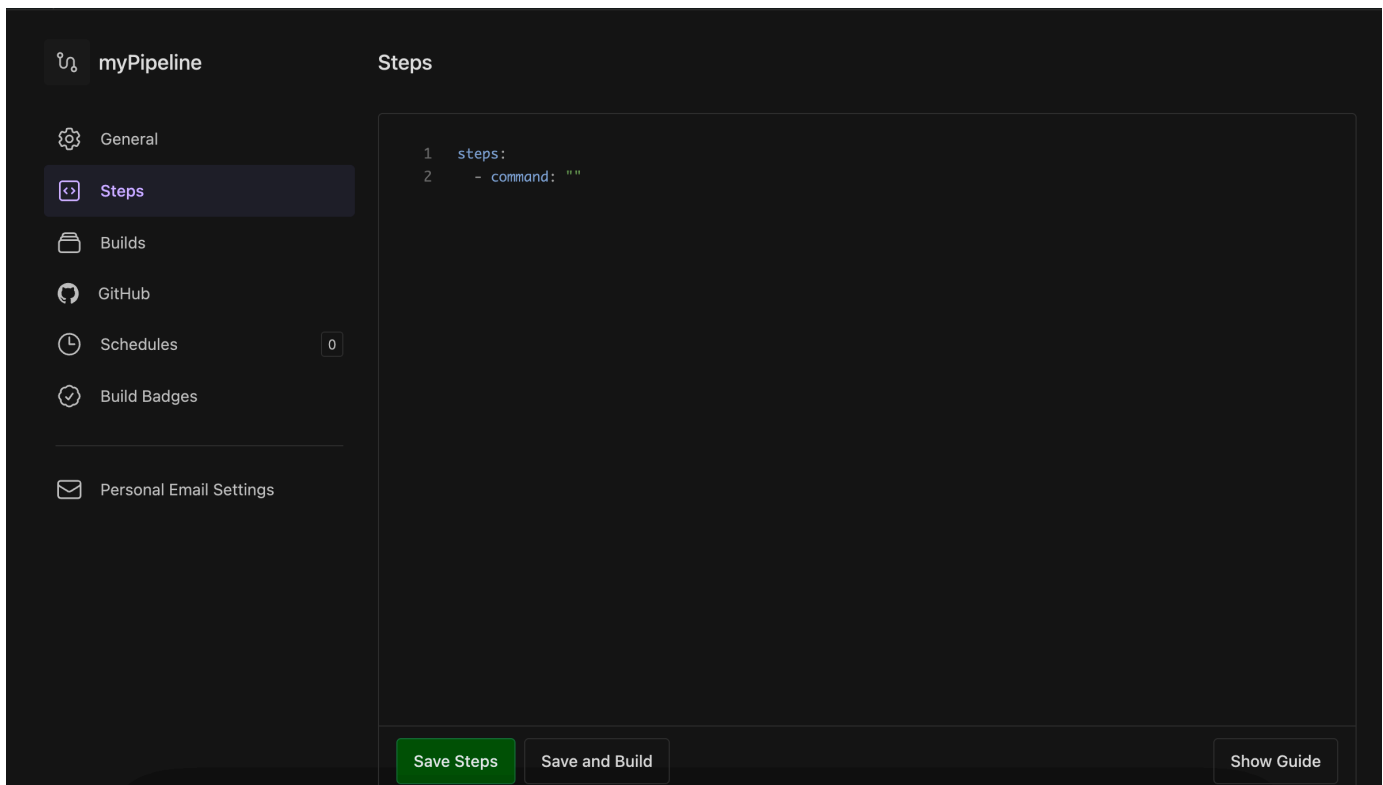
Étape 4 : Mettez à jour les étapes de votre pipeline Buildkite

Au cours de cette étape, vous allez mettre à jour les étapes de votre pipeline Buildkite afin d'ajouter les étiquettes nécessaires et les remplacements facultatifs. Pour obtenir la liste complète des remplacements d'étiquettes pris en charge, consultez [Les remplacements d'étiquettes sont pris en charge avec le moteur CodeBuild Buildkite Runner hébergé](#).

Mettez à jour les étapes de votre pipeline

1. Accédez à la page des étapes de votre pipeline Buildkite en sélectionnant votre pipeline Buildkite, en choisissant Paramètres, puis en choisissant Étapes.

Si ce n'est pas déjà fait, choisissez Convertir en étapes YAML.



2. Au minimum, vous devrez spécifier une [balise d'agent Buildkite](#) faisant référence au nom de votre pipeline. CodeBuild Le nom du projet est nécessaire pour lier les AWS paramètres associés à votre tâche Buildkite à un projet spécifique CodeBuild . En incluant le nom du projet dans le YAML, CodeBuild il est autorisé à invoquer des tâches avec les paramètres de projet corrects.

```
agents:
  project: "codebuild-<project name>"
```

Voici un exemple d'étapes du pipeline Buildkite avec uniquement la balise d'étiquette du projet :

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "echo \"Hello World\""
```

Vous pouvez également remplacer votre image et le type de calcul dans l'étiquette. Consultez [Calculez les images prises en charge avec le moteur CodeBuild Buildkite Runner hébergé](#) la liste des images disponibles. Le type de calcul et l'image figurant dans l'étiquette remplaceront les paramètres d'environnement de votre projet. Pour remplacer les paramètres de votre

environnement pour une version de calcul CodeBuild EC2 ou Lambda, utilisez la syntaxe suivante :

```
agents:
  project: "codebuild-<project name>"
  image: "<environment-type>-<image-identifier>"
  instance-size: "<instance-size>"
```

Voici un exemple d'étapes du pipeline Buildkite avec des remplacements de taille d'image et d'instance :

```
agents:
  project: "codebuild-myProject"
  image: "arm-3.0"
  instance-size: "small"
steps:
  - command: "echo \"Hello World\""
```

Vous pouvez remplacer le parc utilisé pour votre construction dans l'étiquette. Cela remplacera les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour plus d'informations, voir [Exécuter des builds sur des flottes de capacité réservée](#).

Pour remplacer les paramètres de votre flotte pour une version de calcul Amazon EC2, utilisez la syntaxe suivante :

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
```

Pour remplacer à la fois la flotte et l'image utilisées pour la génération, utilisez la syntaxe suivante :

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
  image: "<environment-type>-<image-identifier>"
```

Voici un exemple d'étapes du pipeline Buildkite avec des remplacements de flotte et d'image :

```
agents:
  project: "codebuild-myProject"
  fleet: "myFleet"
  image: "arm-3.0"
steps:
  - command: "echo \"Hello World\""
```

3. Vous pouvez choisir d'exécuter des commandes buildspec en ligne lors de la version auto-hébergée de Buildkite Runner (voir pour plus de détails). [Exécutez les commandes buildspec pour les phases INSTALL, PRE_BUILD et POST_BUILD](#) Pour spécifier que le CodeBuild build doit exécuter les commandes buildspec lors de votre build d'exécution auto-hébergé par Buildkite, utilisez la syntaxe suivante :

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

Voici un exemple de pipeline Buildkite avec une dérogation buildspec :

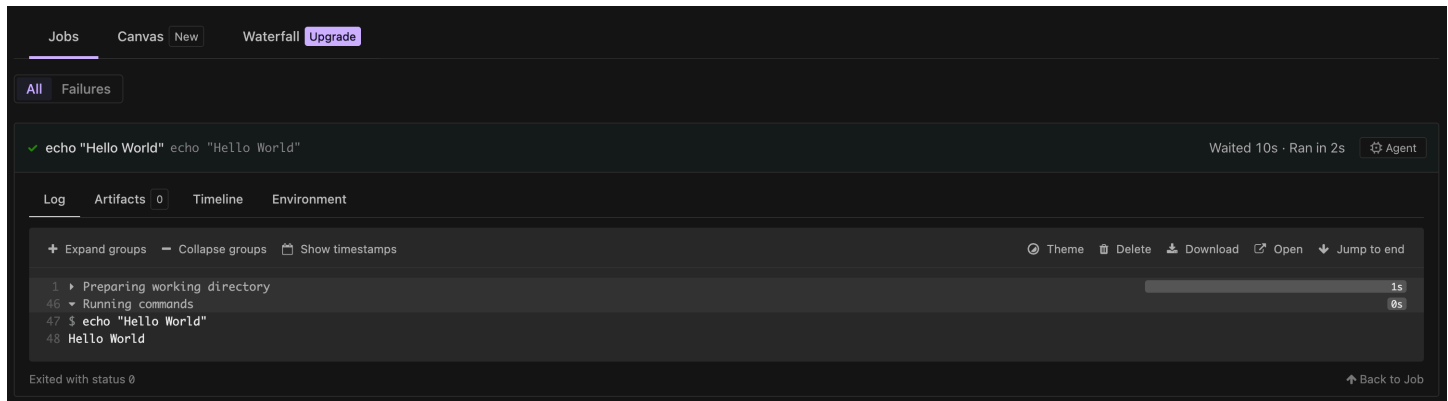
```
agents:
  project: "codebuild-myProject"
  buildspec-override: "true"
steps:
  - command: "echo \"Hello World\""
```

4. Vous pouvez éventuellement fournir des étiquettes autres que celles prises CodeBuild en charge. Ces étiquettes seront ignorées dans le but de remplacer les attributs de la version, mais elles n'échoueront pas à la demande de webhook. Par exemple, l'ajout myLabel : "testLabel" d'une étiquette n'empêchera pas le build de s'exécuter.

Étape 5 : Passez en revue vos résultats

Chaque fois qu'une tâche Buildkite est lancée dans votre pipeline, vous CodeBuild recevrez un événement webhook via le `job.scheduled` webhook Buildkite. Pour chaque tâche de votre build de Buildkite, une version CodeBuild sera lancée pour exécuter un Runner Buildkite éphémère. Le coureur est responsable de l'exécution d'une seule tâche Buildkite. Une fois le travail terminé, le lanceur et le processus de construction associé seront immédiatement interrompus.

Pour consulter les journaux des tâches de votre flux de travail, accédez à votre pipeline Buildkite et sélectionnez la version la plus récente (vous pouvez déclencher une nouvelle version en choisissant Nouvelle version). Une fois que le CodeBuild build associé à chacune de vos tâches démarre et reprend la tâche, vous devriez voir les journaux de la tâche dans la console Buildkite



Authentification de Buildkite auprès d'un dépôt privé

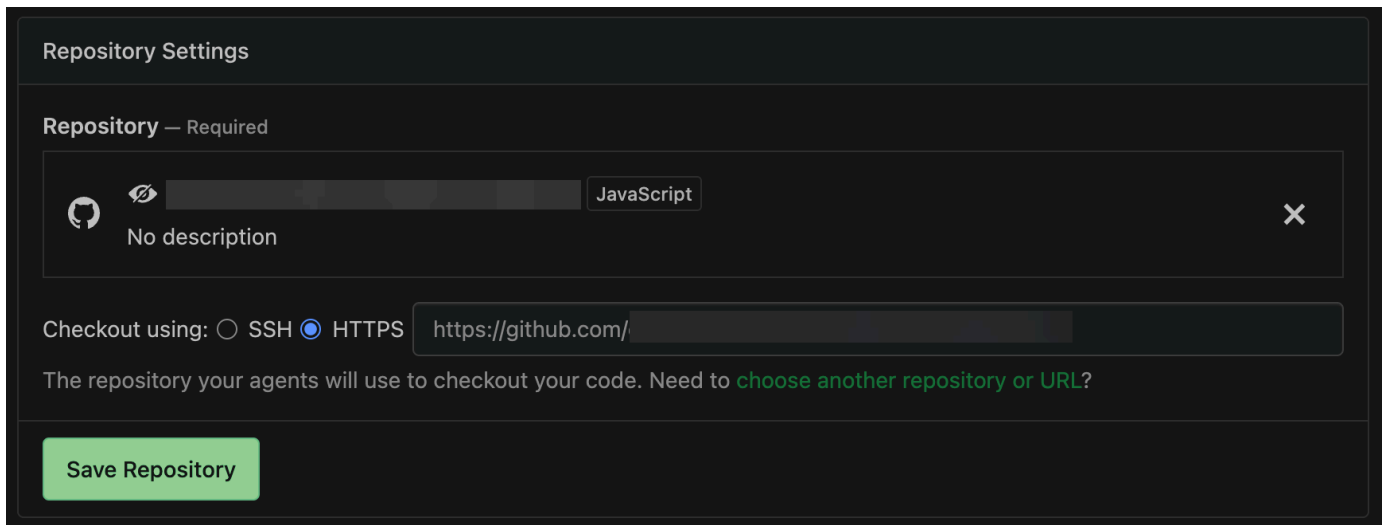
Si vous avez configuré un dépôt privé dans votre pipeline Buildkite, Buildkite a besoin d'[autorisations supplémentaires dans l'environnement de construction pour extraire le](#) référentiel, car Buildkite ne vend pas d'informations d'identification à des exécuteurs auto-hébergés pour les extraire des référentiels privés. Pour authentifier l'agent Runner auto-hébergé par Buildkite auprès de votre référentiel source privé externe, vous pouvez utiliser l'une des options suivantes.

Pour vous authentifier auprès de CodeBuild

CodeBuild propose une gestion des informations d'identification gérées pour les types de sources pris en charge. Pour utiliser les informations d'identification CodeBuild source pour extraire le référentiel source de votre tâche, vous pouvez suivre les étapes suivantes :

1. Dans la CodeBuild console, accédez à Modifier le projet ou créez un nouveau CodeBuild projet en suivant les étapes décrites dans [Étape 2 : Création d'un CodeBuild projet avec un webhook](#).
2. Sous les options d'identification source de Buildkite, sélectionnez le fournisseur du référentiel source de votre tâche.
 1. Si vous souhaitez utiliser les informations d' CodeBuild identification au niveau du compte, vérifiez qu'elles sont correctement configurées. De plus, si une spécification de construction en ligne est configurée dans votre projet, vérifiez qu'elle est activée. [git-credential-helper](#)
 2. Si vous souhaitez utiliser les informations d' CodeBuild identification au niveau du projet, sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement et configurez les informations d'identification pour votre projet.

3. Dans les paramètres de votre pipeline Buildkite, accédez aux paramètres du référentiel. Définissez les paramètres de récupération de votre référentiel source sur Checkout via HTTPS



Repository Settings

Repository — Required

  [Redacted] JavaScript 

No description

Checkout using: ☐ SSH ☒ HTTPS

The repository your agents will use to checkout your code. Need to [choose another repository or URL?](#)

[Save Repository](#)

Pour vous authentifier avec les secrets de Buildkite

Buildkite gère un [plugin ssh-checkout](#) qui peut être utilisé pour authentifier le runner auto-hébergé auprès d'un référentiel source externe à l'aide d'une clé ssh. La valeur clé est stockée sous forme de [secret Buildkite](#) et récupérée automatiquement par l'agent Runner auto-hébergé par Buildkite lorsqu'il tente d'extraire un dépôt privé. Afin de configurer le plugin ssh-checkout pour votre pipeline Buildkite, vous pouvez suivre les étapes suivantes :

1. Générez une clé SSH privée et publique en utilisant votre adresse e-mail, par exemple `ssh-keygen -t rsa -b 4096 -C "myEmail@address.com"`
2. Ajoutez la clé publique à votre référentiel source privé. Par exemple, vous pouvez suivre [ce guide](#) pour ajouter une clé à un GitHub compte.
3. Ajoutez une [nouvelle clé secrète SSH](#) à votre cluster Buildkite. Dans votre cluster Buildkite, sélectionnez Secrets → Nouveau secret. Donnez un nom à votre secret dans le champ Clé et ajoutez votre clé SSH privée dans le champ Valeur :

New Secret

Key — Required

SOURCE_SSH_KEY

Keys are case insensitive, can only contain alphanumeric and underscore characters, and can't start with BUILDKITE or BK

Value — Required

-----BEGIN OPENSSH PRIVATE KEY-----

4. Dans votre pipeline Buildkite, accédez aux paramètres de votre dépôt et configurez le checkout pour qu'il utilise SSH.

Repository Settings

Repository — Required

  [Redacted] 
No description

Checkout using: ☒ SSH ☐ HTTPS

The repository your agents will use to checkout your code. Need to [choose another repository or URL?](#)

[Save Repository](#)

5. Mettez à jour les étapes YAML de votre pipeline pour utiliser le `git-ssh-checkout` plugin. Par exemple, le fichier YAML de pipeline suivant utilise l'action de paiement avec la clé secrète Buildkite ci-dessus :

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "npm run build"
    plugins:
      - git-ssh-checkout#v0.4.1:
```



```
ssh-secret-key-name: 'SOURCE_SSH_KEY'
```

6. Lorsque vous exécutez une tâche d'exécution auto-hébergée par Buildkite dans Buildkite CodeBuild, Buildkite utilise désormais automatiquement votre valeur secrète configurée lors de l'extraction de votre dépôt privé

Options de configuration du coureur

Vous pouvez spécifier les variables d'environnement suivantes dans la configuration de votre projet pour modifier la configuration de vos coureurs auto-hébergés :

- `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`: CodeBuild récupérera la valeur secrète configurée comme valeur de cette variable d'environnement afin d'enregistrer l' AWS Secrets Manager agent d'exécution auto-hébergé par Buildkite. Cette variable d'environnement doit être de type `SECRETS_MANAGER` et sa valeur doit être le nom de votre secret dans Secrets Manager. Une variable d'environnement de jeton d'agent Buildkite est requise pour tous les projets Buildkite Runner.
- `CODEBUILD_CONFIG_BUILDKITE_CREDENTIAL_DISABLE`: Par défaut, CodeBuild chargera les informations d'identification source au niveau du compte ou du projet dans l'environnement de construction, car ces informations d'identification sont utilisées par l'agent Buildkite pour extraire le référentiel source de la tâche. Pour désactiver ce comportement, vous pouvez ajouter cette variable d'environnement à votre projet avec la valeur définie sur `true`, ce qui empêchera le chargement des informations d'identification de la source dans l'environnement de génération.

Exécutez les commandes buildspec pour les phases `INSTALL`, `PRE_BUILD` et `POST_BUILD`

Par défaut, CodeBuild ignore les commandes buildspec lors de l'exécution d'une version auto-hébergée de Buildkite Runner. Pour exécuter les commandes buildspec pendant la construction,

```
buildspec-override: "true"
```

peut être ajouté en tant que suffixe à l'étiquette :

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

En utilisant cette commande, CodeBuild vous créez un dossier appelé `buildkite-runner` dans le dossier source principal du conteneur. Lorsque le lanceur Buildkite démarre pendant la BUILD phase, il s'exécute dans le `buildkite-runner` répertoire.

L'utilisation d'un override buildspec dans une version Buildkite auto-hébergée présente plusieurs limites :

- L'agent Buildkite nécessite que les informations d'identification de la source existent dans l'environnement de construction pour extraire le référentiel source de la tâche. Si vous utilisez les informations d'identification de la CodeBuild source pour l'authentification, vous devez les activer `git-credential-helper` dans votre buildspec. Par exemple, vous pouvez utiliser les spécifications de construction suivantes `git-credential-helper` pour activer vos versions de Buildkite :

```
version: 0.2
env:
  git-credential-helper: yes
phases:
  pre_build:
    commands:
      - echo "Hello World"
```

- CodeBuild n'exécutera pas de commandes buildspec pendant la BUILD phase, car le lanceur auto-hébergé s'exécute pendant la phase. BUILD
- CodeBuild ne prend pas en charge les fichiers buildspec pour les versions de Buildkite Runner. Seules les spécifications de construction en ligne sont prises en charge pour les coureurs auto-hébergés par Buildkite
- Si une commande de construction échoue pendant la INSTALL phase PRE_BUILD ou, elle ne CodeBuild démarrera pas le lanceur auto-hébergé et la tâche Buildkite devra être annulée manuellement.

Configuration d'un Buildkite Runner par programmation

Pour configurer un projet Buildkite Runner par programmation, vous devez configurer les ressources suivantes :

Pour créer un Runner Buildkite par programmation

1. Créez un jeton d'agent Buildkite et enregistrez-le en texte clair à l'intérieur.
AWS Secrets Manager
2. Configurez un CodeBuild projet avec la configuration de votre choix. Vous devrez configurer les attributs supplémentaires suivants :
 1. Une valeur d'environnement dont le nom `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`, le type `SECRETS_MANAGER` et une valeur sont égaux au jeton d'agent Buildkite associé à votre cluster Buildkite.
 2. Type de source égal à `NO_SOURCE`
 3. Autorisations d'accès au secret créé à l'étape 1 dans le rôle de service de votre projet

Par exemple, vous pouvez utiliser la commande suivante pour créer un projet Buildkite Runner valide via la CLI :

```
aws codebuild create-project \
--name buildkite-runner-project \
--source "{\"type\": \"NO_SOURCE\", \"buildspec\": \"\"}" \
--environment "{\"image\": \"aws/codebuild/amazonlinux-x86_64-standard:5.0\", \"type\": \"LINUX_CONTAINER\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\", \"environmentVariables\": [{\"name\": \"CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN\", \"type\": \"SECRETS_MANAGER\", \"value\": \"<buildkite-secret-name>\"}]}" \
--artifacts "{\"type\": \"NO_ARTIFACTS\"}" \
--service-role <service-role>
```

3. Créez un webhook Buildkite Runner sur le projet créé à l'étape 2. Vous devrez utiliser les options de configuration suivantes lors de la création du webhook :
 1. build-type doit être égal à `RUNNER_BUILDKITE_BUILD`
 2. Un filtre dont le type `EVENT` et le motif sont égaux à `WORKFLOW_JOB_QUEUED`

Par exemple, vous pouvez utiliser la commande suivante pour créer un webhook Buildkite Runner valide via la CLI :

```
aws codebuild create-webhook \
--project-name buildkite-runner-project \
--filter-groups "[[{"type\": \"EVENT\", \"pattern\": \"WORKFLOW_JOB_QUEUED\"}]]" \
```

```
--build-type RUNNER_BUILDKITE_BUILD
```

4. Enregistrez l'URL de charge utile et les valeurs secrètes renvoyées par l'`create-webhook` et utilisez les informations d'identification pour créer un webhook dans la console Buildkite. Vous pouvez consulter l'étape 3 : Créer un CodeBuild webhook dans Buildkite [Tutoriel : Configuration d'un CodeBuild moteur Buildkite hébergé](#) pour obtenir un guide sur la façon de configurer cette ressource.

Résoudre les problèmes liés au webhook en cas d'échec des builds ou d'une tâche bloquée

Problème:

Le webhook que vous avez configuré ne fonctionne [Tutoriel : Configuration d'un CodeBuild moteur Buildkite hébergé](#) pas ou votre tâche de flux de travail est bloquée dans Buildkite.

Causes possibles :

- Votre événement webhook `job.scheduled` ne parvient peut-être pas à déclencher une compilation. Consultez les journaux des réponses pour voir la réponse ou le message d'erreur.
- Votre CodeBuild build échoue avant que l'agent de course auto-hébergé par Buildkite ne démarre pour s'occuper de votre travail.

Solutions recommandées :

Pour déboguer les événements du webhook Buildkite ayant échoué, procédez comme suit :

1. Dans les paramètres de votre organisation Buildkite, accédez aux services de notification, sélectionnez votre CodeBuild webhook, puis recherchez le journal des demandes.
2. Trouvez l'événement `job.scheduled` webhook associé à votre tâche Buildkite bloquée. Vous pouvez utiliser le champ ID de tâche dans la charge utile du webhook pour corréler l'événement du webhook à votre tâche Buildkite.
3. Sélectionnez l'onglet Réponse et vérifiez le corps de la réponse. Vérifiez que le code d'état de la réponse est 200 valide et que le corps de la réponse ne contient aucun message inattendu.

The screenshot displays the 'Request Log' interface in AWS CodeBuild. At the top, a red box indicates a '400' status code. To its right, the request is identified as 'job.scheduled' with a duration of '0.29s' and a timestamp of '2024-12-11 23:03:33 UTC'. Below this, there are tabs for 'Request' and 'Response', with 'Response' being the active tab. Under the 'Response' tab, the 'Headers' section lists several key-value pairs: 'Date: Wed, 11 Dec 2024 23:03:33 GMT', 'Connection: close', 'Content-Type: application/json', 'Content-Length: 92', 'X-Amzn-Errortype: InvalidInputException:http://', and 'X-Amzn-Requestid: 7a931bed-0bae-4c1a-9f5b-2178900cd180'. The 'Body' section shows a JSON object with a single key 'message' containing the text: 'Project name in label nonMatchingProjectName did not match actual project name'.

Résoudre les problèmes d'autorisation du webhook

Problème:

La tâche Buildkite ne parvient pas à extraire le référentiel source de la tâche en raison de problèmes d'autorisation.

Causes possibles :

- CodeBuild ne dispose pas des autorisations suffisantes pour extraire le référentiel source de la tâche.
- Les paramètres du référentiel du pipeline sont définis pour être extraits à l'aide de SSH pour les informations d'identification CodeBuild gérées.

Solutions recommandées :

- Vérifiez que CodeBuild les autorisations configurées sont suffisantes pour consulter le référentiel source de la tâche. Vérifiez également que le rôle de service de votre CodeBuild projet dispose d'autorisations suffisantes pour accéder à l'option d'autorisation source configurée.
- Vérifiez que votre pipeline Buildkite est configuré pour utiliser le paiement via HTTPS si vous utilisez les informations d'identification du référentiel source CodeBuild géré.

Les remplacements d'étiquettes sont pris en charge avec le moteur CodeBuild Buildkite Runner hébergé

Dans les étiquettes d'agent de votre pipeline Buildkite Steps, vous pouvez fournir une variété de remplacements d'étiquettes qui modifient votre version de course auto-hébergée. Toutes les versions non reconnues par CodeBuild seront ignorées mais n'échoueront pas à votre demande de webhook. Par exemple, le flux de travail YAML suivant inclut des remplacements pour l'image, la taille de l'instance, le parc et les spécifications de construction :

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
      buildspec-override: "true"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "a12-5.0"
        size:
          - "small"
          - "large"
```

project:codebuild-*<project-name>* (obligatoire)

- Exemple : project: "codebuild-myProject"
- Nécessaire pour toutes les configurations d'étapes du pipeline Buildkite. *<project name>* doit être égal au nom du projet pour lequel le runner webhook auto-hébergé est configuré.

queue: "*<queue-name>*"

- Exemple : queue: "*<queue-name>*"
- Utilisé pour acheminer les tâches Buildkite vers une file d'attente spécifique. Consultez le [tag Buildkite Agent Queue](#) pour plus d'informations.

image: "<environment-type>-<image-identifiant>"

- Exemple : image: "arm-3.0"
- Remplace l'image et le type d'environnement utilisés lors du démarrage de la version auto-hébergée par Runner avec une image organisée. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculez les images prises en charge avec le moteur CodeBuild Buildkite Runner hébergé](#).

1. Pour remplacer l'image et le type d'environnement utilisés par une image personnalisée, utilisez
image: "custom-<environment-type>-<custom-image-identifiant>"

2. Exemple :

```
image:
  "custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0"
```

 Note

Si l'image personnalisée réside dans un registre privé, vous devez configurer les informations d'identification de registre appropriées dans votre CodeBuild projet.

instance-size: "<instance-size>"

- Exemple : instance-size: "medium"
- Remplace le type d'instance utilisé lors du démarrage de la version de course auto-hébergée. Pour en savoir plus sur les valeurs prises en charge, consultez [Calculez les images prises en charge avec le moteur CodeBuild Buildkite Runner hébergé](#).

fleet: "<fleet-name>"

- Exemple : fleet: "myFleet"
- Remplace les paramètres de flotte configurés dans votre projet pour utiliser le parc spécifié. Pour plus d'informations, voir [Exécuter des builds sur des flottes de capacité réservée](#).

buildspec-override: "<boolean>"

- Exemple : buildspec-override: "true"

- Permet à la compilation d'exécuter des commandes buildspec dans les POST_BUILD phases INSTALLPRE_BUILD, et si elle est définie sur. true

Calculez les images prises en charge avec le moteur CodeBuild Buildkite Runner hébergé

Dans l'étiquette que vous avez configurée [Buildkite Runner autogéré dans AWS CodeBuild](#), vous pouvez remplacer les paramètres de votre environnement Amazon EC2 en utilisant les valeurs des trois premières colonnes. CodeBuild fournit les images de calcul Amazon EC2 suivantes. Pour plus d'informations sur

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
		gpu_small			
		gpu_large			
linux-ec2	latest	small	Amazon Linux 2	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Aucune
		medium			
		large			

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2,0
arm	3.0	2xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3,0
arm-ec2	latest	small medium large	Amazon Linux 2	aws/codebuild/ami/amazonlinux-arm-base:latest	Aucune
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	Ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	Ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	Ubuntu/standard/7.0

Type d'environnement	Identificateur de l'image	Taille d'instance	Plateforme	Image résolue	Définition
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/A
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Aucune

En outre, vous pouvez modifier les paramètres de votre environnement Lambda en utilisant les valeurs suivantes. Pour plus d'informations sur le calcul CodeBuild Lambda, consultez. [Exécuter des builds sur AWS Lambda ordinateur](#) CodeBuild prend en charge les images de calcul Lambda suivantes :

Type d'environnement	Identificateur de l'image	Taille d'instance			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
	corretto1	4GB			
	1	8GB			
	corretto1	10GB			
	7				
	corretto2				
	1				
	nodejs18				
	nodejs20				
arm-lambda	python3.1				
	1				
	python3.1				
	2				
	ruby3.2				

Pour plus d'informations, consultez [Modes et types de calcul de l'environnement de création](#) et [Images Docker fournies par CodeBuild](#).

Utilisez des webhooks avec AWS CodeBuild

AWS CodeBuild prend en charge l'intégration de webhook avec GitHub GitHub Enterprise Server GitLab, GitLab Self Managed et Bitbucket.

Rubriques

- [Bonnes pratiques d'utilisation des webhooks avec AWS CodeBuild](#)

- [Événements du webhook Bitbucket](#)
- [GitHub webhooks mondiaux et organisationnels](#)
- [GitHub webhooks manuels](#)
- [GitHub événements webhook](#)
- [GitLab webhooks de groupe](#)
- [GitLab webhooks manuels](#)
- [GitLab événements webhook](#)
- [Webhooks manuels Buildkite](#)
- [Approbation des commentaires par pull request](#)

Bonnes pratiques d'utilisation des webhooks avec AWS CodeBuild

Pour les projets qui utilisent des référentiels publics pour configurer des webhooks, nous recommandons les options suivantes :

ACTOR_ACCOUNT_ID Filtres de configuration

Ajoutez des ACTOR_ACCOUNT_ID filtres aux groupes de filtres Webhook de votre projet pour spécifier quels utilisateurs peuvent déclencher une génération. Chaque événement webhook CodeBuild livré à est accompagné d'informations sur l'expéditeur qui spécifient l'identifiant de l'acteur. CodeBuild filtrera les webhooks en fonction du modèle d'expression régulière fourni dans les filtres. Vous pouvez spécifier les utilisateurs spécifiques autorisés à déclencher des builds avec ce filtre. Pour plus d'informations, consultez [GitHub événements webhook](#) et [Événements du webhook Bitbucket](#).

FILE_PATH Filtres de configuration

Ajoutez des FILE_PATH filtres aux groupes de filtres Webhook de votre projet pour inclure ou exclure les fichiers susceptibles de déclencher une génération en cas de modification. Par exemple, vous pouvez refuser les demandes de compilation visant à modifier le `buildspec.yml` fichier à l'aide d'un modèle d'expression régulière tel que `^buildspec.yml$`, avec la `excludeMatchedPattern` propriété. Pour plus d'informations, consultez [GitHub événements webhook](#) et [Événements du webhook Bitbucket](#).

Limitez les autorisations associées à votre rôle IAM de build

Les builds déclenchés par un webhook utilisent le rôle de service IAM spécifié dans le projet. Nous vous recommandons de définir les autorisations du rôle de service sur l'ensemble minimal

d'autorisations requis pour exécuter la génération. Par exemple, dans un scénario de test et de déploiement, créez un projet pour les tests et un autre pour le déploiement. Le projet de test accepte les versions de webhook à partir du référentiel, mais ne fournit aucune autorisation d'écriture sur vos ressources. Le projet de déploiement fournit des autorisations d'écriture à vos ressources, et le filtre Webhook est configuré pour autoriser uniquement les utilisateurs de confiance à déclencher des builds.

Utilisez une spécification de construction en ligne ou stockée sur Amazon S3

Si vous définissez votre buildspec en ligne dans le projet lui-même, ou si vous stockez le fichier buildspec dans un compartiment Amazon S3, le fichier buildspec n'est visible que par le propriétaire du projet. Cela empêche les pull requests d'apporter des modifications de code au fichier buildspec et de déclencher des builds indésirables. Pour plus d'informations, consultez le [ProjectSourcefichier .buildspec](#) dans la référence de l'API. CodeBuild

Événements du webhook Bitbucket

Vous pouvez utiliser des groupes de filtres webhook pour spécifier quels événements webhook Bitbucket déclenchent une génération. Par exemple, vous pouvez spécifier qu'une construction n'est déclenchée que pour les modifications apportées à des branches spécifiques.

Vous pouvez créer un ou plusieurs des groupes de filtres webhook pour spécifier les événements webhook qui déclenchent une génération. Une génération est déclenchée si un groupe de filtres est évalué comme vrai, ce qui se produit lorsque tous les filtres du groupe sont évalués comme vrais. Lorsque vous créez un groupe de filtres, vous spécifiez :

Un événement

Pour Bitbucket, vous pouvez choisir un ou plusieurs des événements suivants :

- PUSH
- PULL_REQUEST_CREATED
- PULL_REQUEST_UPDATED
- PULL_REQUEST_MERGED
- PULL_REQUEST_CLOSED

Le type d'événement du webhook est situé dans son en-tête dans le champ X-Event-Key. Le tableau suivant présente le mappage des valeurs d'en-tête X-Event-Key aux types d'événements.

 Note

Vous devez activer l'événement `merged` dans le paramètre de votre webhook Bitbucket si vous créez un groupe de filtres de webhook qui utilise le type d'événement `PULL_REQUEST_MERGED`. Vous devez également activer l'événement `declined` dans vos paramètres de webhook Bitbucket si vous créez un groupe de filtres de webhook utilisant le `PULL_REQUEST_CLOSED` type d'événement.

Valeur d'en-tête X-Event-Key	Type d'événement
<code>repo:push</code>	<code>PUSH</code>
<code>pullrequest:created</code>	<code>PULL_REQUEST_CREATED</code>
<code>pullrequest:updated</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pullrequest:fulfilled</code>	<code>PULL_REQUEST_MERGED</code>
<code>pullrequest:rejected</code>	<code>PULL_REQUEST_CLOSED</code>

En `PULL_REQUEST_MERGED` effet, si une pull request est fusionnée avec la stratégie squash et que la branche pull request est fermée, le commit de la pull request d'origine n'existe plus. Dans ce cas, la variable d'environnement `CODEBUILD_WEBHOOK_MERGE_COMMIT` contient l'identifiant du commit de fusion écrasé.

Un ou plusieurs filtres optionnels

Utilisez une expression régulière pour spécifier un filtre. Pour qu'un événement déclenche une génération, chaque filtre du groupe qui lui est associé doit avoir la valeur `true`.

`ACTOR_ACCOUNT_ID(ACTOR_ID dans la console)`

Un événement webhook déclenche une compilation lorsqu'un identifiant de compte Bitbucket correspond au modèle d'expression régulière. Cette valeur est disponible dans la propriété `account_id` de l'objet `actor` dans la charge utile de filtre du webhook.

`HEAD_REF`

Un événement webhook déclenche une construction lorsque la référence principale correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name`

etrefs/tags/tag-name). Un filtre HEAD_REF évalue le nom de la référence Git de la branche ou de la balise. Le nom de la branche ou de la balise est situé dans le champ name de l'objet new dans l'objet push de la charge utile du webhook. Pour des événements de demande d'extraction, le nom de la branche est situé dans le champ name de l'objet branch de l'objet source dans la charge utile du webhook.

BASE_REF

Un événement webhook déclenche une construction lorsque la référence de base correspond au modèle d'expression régulière. Un filtre BASE_REF fonctionne avec des événements de demande d'extraction uniquement (par exemple, refs/heads/branch-name). Un filtre BASE_REF évalue le nom de la référence Git de la branche. Le nom de la branche est situé dans le champ name de l'objet branch dans l'objet destination de la charge utile du webhook.

FILE_PATH

Un webhook déclenche une compilation lorsque le chemin d'un fichier modifié correspond au modèle d'expression régulière.

COMMIT_MESSAGE

Un webhook déclenche une compilation lorsque le message de validation principal correspond au modèle d'expression régulière.

WORKFLOW_NAME

Un webhook déclenche une compilation lorsque le nom du flux de travail correspond au modèle d'expression régulière.

Note

Vous pouvez trouver la charge utile du webhook dans les paramètres webhook de votre référentiel Bitbucket.

Rubriques

- [Filtrage des événements webhook BitBucket \(console\)](#)
- [Filtrage des événements webhook BitBucket \(kit SDK\)](#)
- [Filtrage des événements webhook Bitbucket \(CloudFormation\)](#)

Filtrage des événements webhook BitBucket (console)

Pour utiliser le AWS Management Console pour filtrer les événements du webhook :

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Add filter group (Ajouter un groupe de filtres) pour ajouter un autre groupe de filtres.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API AWS CodeBuild .

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour des demandes d'extraction uniquement :

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

PULL_REQUEST_CLOSED ✕

► Start a build under these conditions - *optional*

► Don't start a build under these conditions - *optional*

Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur true :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/branch1!`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour toutes les demandes à l'exception des événements de balise.

Filter group 1

Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

► Start a build under these conditions - optional

▼ Don't start a build under these conditions - optional

Add filter

Filter 1

Type

HEAD_REF ▼

Pattern

^refs/tags/.*

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur Bitbucket qui ne dispose pas d'un ID de compte correspondant à l'expression régulière `actor-account-id`.

 Note

Pour savoir comment trouver l'identifiant de votre compte Bitbucket, consultez <https://api.bitbucket.org/2.0/users/user-name>, où se *user-name* trouve votre nom d'utilisateur Bitbucket.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

PULL_REQUEST_CLOSED ✕

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour un événement de transmission lorsque le message de validation principal correspond à l'expression régulière `\[CodeBuild\]`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - *optional*

HEAD_REF - *optional*

BASE_REF - *optional*

FILE_PATH - *optional*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Filtrage des événements webhook BitBucket (kit SDK)

Pour utiliser le AWS CodeBuild SDK afin de filtrer les événements Webhook, utilisez le `filterGroups` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Pour créer un filtre webhook qui déclenche une génération pour des demandes d'extraction uniquement, insérez les éléments suivants dans la syntaxe de demande :

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    }
  ]
]
```

Pour créer un filtre webhook qui déclenche une génération pour des branches spécifiées uniquement, utilisez le paramètre `pattern` pour spécifier une expression régulière pour filtrer les noms de

branche. Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur true :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/myBranch$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Vous pouvez utiliser le paramètre `excludeMatchedPattern` pour spécifier les événements qui ne déclenchent pas une génération. Dans cet exemple, une génération est déclenchée pour toutes les demandes à l'exception des événements de balise.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur Bitbucket avec l'ID de compte `actor-account-id`.

Note

Pour savoir comment trouver l'identifiant de votre compte Bitbucket, consultez <https://api.bitbucket.org/2.0/users/user-name>, où se *user-name* trouve votre nom d'utilisateur Bitbucket.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",  
      "pattern": "actor-account-id"  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière dans l'argument `pattern` changent. Dans cet exemple,

le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^buildspec.*"  
    }  
  ]  
]
```

Dans cet exemple, le groupe de filtres indique qu'une compilation est déclenchée uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^src/.+|^test/.+"  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque le message de validation principal correspond à l'expression régulière de l'argument de schéma. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque le message de validation principal de l'événement de transmission correspond à l'expression régulière `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  

```

```
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "COMMIT_MESSAGE",
    "pattern": "\\[CodeBuild\\]"
  }
]
```

Filtrage des événements webhook Bitbucket (CloudFormation)

Pour utiliser un CloudFormation modèle afin de filtrer les événements Webhook, utilisez la `FilterGroups` propriété du AWS CodeBuild projet. La partie au format YAML suivante d'un modèle CloudFormation crée deux groupes de filtres. Ensemble, ils déclenchent une génération lorsque l'un des deux ou les deux prennent la valeur `true` :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` par un utilisateur Bitbucket n'ayant pas d'ID de compte 12345.
- Le deuxième groupe de filtres spécifie des demandes de transmission créées sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\\[CodeBuild\\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
```

```
Webhook: true
FilterGroups:
- - Type: EVENT
  Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
- - Type: BASE_REF
  Pattern: ^refs/heads/main$
  ExcludeMatchedPattern: false
- - Type: ACTOR_ACCOUNT_ID
  Pattern: 12345
  ExcludeMatchedPattern: true
- - Type: EVENT
  Pattern: PUSH
- - Type: HEAD_REF
  Pattern: ^refs/heads/.+
- - Type: FILE_PATH
  Pattern: READ_ME
  ExcludeMatchedPattern: true
- - Type: EVENT
  Pattern: PUSH
- - Type: COMMIT_MESSAGE
  Pattern: \[CodeBuild\]
- - Type: FILE_PATH
  Pattern: ^src/.+|^test/.+
```

GitHub webhooks mondiaux et organisationnels

Vous pouvez utiliser des webhooks CodeBuild GitHub globaux ou organisationnels pour démarrer des builds à partir d'événements webhooks à partir de n'importe quel référentiel au sein d'une GitHub organisation ou d'une entreprise. Les webhooks globaux et organisationnels fonctionnent avec tous les types d'événements GitHub webhook existants et peuvent être configurés en ajoutant une configuration de portée lors de la création d'un CodeBuild webhook. Vous pouvez également utiliser des webhooks mondiaux et organisationnels pour [configurer des GitHub Action Runners auto-hébergés au sein d'un même projet CodeBuild](#) afin de recevoir des WORKFLOW_JOB_QUEUED événements provenant de plusieurs référentiels au sein d'un même projet.

Rubriques

- [Configurer un GitHub webhook mondial ou organisationnel](#)
- [Filtrer les événements webhook GitHub mondiaux ou organisationnels \(console\)](#)
- [Filtrer les événements du webhook de l' GitHub organisation \(CloudFormation\)](#)

Configurer un GitHub webhook mondial ou organisationnel

Les étapes de haut niveau pour configurer un GitHub webhook mondial ou organisationnel sont les suivantes. Pour plus d'informations sur les GitHub webhooks mondiaux et organisationnels, consultez [GitHub webhooks mondiaux et organisationnels](#).

1. Définissez l'emplacement source de votre projet sur `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. Dans la configuration de la portée du webhook, définissez la portée sur l'une ou l'autre `GITHUB_ORGANIZATION` ou `GITHUB_GLOBAL` selon qu'il doit s'agir d'une organisation ou d'un [webhook mondial](#). Pour plus d'informations, consultez la section [Types de webhooks](#).
3. Spécifiez un nom dans le cadre de la configuration du scope du webhook. Pour les webhooks d'organisation, il s'agit du nom de l'organisation, et pour les webhooks globaux, il s'agit du nom de l'entreprise.

Note

Si le type de source du projet est `GITHUB_ENTERPRISE`, vous devrez également spécifier un domaine dans le cadre de la configuration de la portée du webhook.

4. (Facultatif) Si vous souhaitez uniquement recevoir des événements de webhook pour des référentiels spécifiques au sein de votre organisation ou de votre entreprise, vous pouvez les spécifier `REPOSITORY_NAME` sous forme de filtre lors de la création du webhook.
5. Si vous créez un webhook d'organisation, assurez-vous qu'il CodeBuild dispose des autorisations nécessaires pour créer des webhooks au niveau de l'organisation. GitHub Vous pouvez créer un jeton d'accès GitHub personnel avec les autorisations Webhook de l'organisation, ou l'utiliser CodeBuild OAuth. Pour de plus amples informations, veuillez consulter [GitHub et jeton d'accès au serveur GitHub Enterprise](#).

Notez que les webhooks d'organisation fonctionnent avec tous les types d'événements GitHub webhook existants.

6. Si vous créez un webhook global, le webhook devra être créé manuellement. Pour plus d'informations sur la création manuelle d'un webhook à l'intérieur de celui-ci GitHub, consultez [GitHub webhooks manuels](#).

Notez que les webhooks globaux ne prennent en charge que le type `WORKFLOW_JOB_QUEUED` d'événement. Pour de plus amples informations, veuillez consulter [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

Filtrer les événements webhook GitHub mondiaux ou organisationnels (console)

Lorsque vous créez un GitHub projet via la console, sélectionnez les options suivantes pour créer un webhook GitHub global ou organisationnel au sein du projet. Pour plus d'informations sur les GitHub webhooks mondiaux et organisationnels, consultez [GitHub webhooks mondiaux et organisationnels](#).

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Source:
 - Dans le champ Source provider, sélectionnez GitHubGitHubEnterprise.
 - Pour Repository, choisissez un GitHubwebhook scopé.

Le GitHub référentiel sera automatiquement défini sur `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, qui est l'emplacement source requis pour les webhooks mondiaux et organisationnels.

Note

Si vous utilisez des webhooks d'organisation, assurez-vous que celle-ci CodeBuild dispose des autorisations nécessaires pour créer des webhooks au niveau de l'organisation. GitHub Si vous utilisez une [OAuthconnexion existante](#), il se peut que vous deviez la régénérer pour accorder CodeBuild cette autorisation. Vous pouvez également créer le webhook manuellement à l'aide de la fonction [webhooks CodeBuild manuels](#). Notez que si vous avez un GitHub OAuth jeton existant et que vous souhaitez ajouter des autorisations d'organisation supplémentaires, vous pouvez [révoquer l'autorisation du OAuth jeton](#) et le reconnecter via la CodeBuild console.

Source**Add source****Source 1 - Primary**

Source provider

GitHub ▼

Repository

☐ Repository in my GitHub account☐ Public repository☒ GitHub scoped webhook

GitHub repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

Connection status

You are connected to GitHub using a personal access token.

Disconnect from GitHub

- Dans les événements Webhook de la source principale :
 - Pour Type de portée, choisissez Niveau organisation si vous créez un webhook d'organisation ou Niveau entreprise si vous créez un webhook global.
 - Dans Nom, entrez le nom de l'entreprise ou de l'organisation, selon qu'il s'agit d'un webhook mondial ou d'une organisation.

Si le type de source du projet est `GITHUB_ENTERPRISE`, vous devez également spécifier un domaine dans le cadre de la configuration de l'organisation du webhook. Par exemple, si l'URL de votre organisation est `https://domain.com/orgs/org-name`, le domaine est `https://domain.com`.

Note

Ce nom ne peut pas être modifié une fois le webhook créé. Pour modifier le nom, vous pouvez supprimer et recréer le webhook. Si vous souhaitez supprimer complètement le webhook, vous pouvez également mettre à jour l'emplacement de la source du projet vers un GitHub référentiel.

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#) ☒ Rebuild every time a code change is pushed to this repository

Scope type

☒ Organization level☐ Enterprise level

Organization name

Your GitHub organization name.

Build type

☒ Single build
Triggers single build☐ Batch build
Triggers multiple builds as single execution**► Additional configuration**

- (Facultatif) Dans les groupes de filtres d'événements Webhook, vous pouvez spécifier [les événements pour lesquels vous souhaitez déclencher une nouvelle génération](#). Vous pouvez également spécifier `REPOSITORY_NAME` comme filtre de manière à ne déclencher des builds que sur des événements webhook provenant de référentiels spécifiques.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Vous pouvez également définir le type d'événement sur WORKFLOW_JOB_QUEUED pour configurer des coureurs GitHub Actions auto-hébergés. Pour de plus amples informations, veuillez consulter [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

3. Continuez avec les valeurs par défaut, puis choisissez Create build project.

Filtrer les événements du webhook de l' GitHub organisation ()CloudFormation

Pour utiliser un CloudFormation modèle afin de filtrer les événements webhook d'une organisation, utilisez la ScopeConfiguration propriété du AWS CodeBuild projet. Pour plus d'informations sur les GitHub webhooks mondiaux et organisationnels, consultez [GitHub webhooks mondiaux et organisationnels](#).

Note

Les webhooks globaux et les webhooks GitHub d'entreprise ne sont pas pris en charge par CloudFormation

La partie suivante d'un CloudFormation modèle au format YAML crée quatre groupes de filtres. Ensemble, ils déclenchent une compilation lorsqu'une ou toutes les évaluations sont vraies :

- Le premier groupe de filtres indique que les pull requests sont créées ou mises à jour sur les branches dont les noms de référence Git correspondent à l'expression régulière `^refs/heads/main$` d'un GitHub utilisateur qui n'a pas d'identifiant de compte12345.
- Le deuxième groupe de filtres spécifie les demandes de transmission créées sur des fichiers dont les noms correspondent à l'expression régulière `README` dans des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\[CodeBuild\]`.
- Le quatrième groupe de filtres spécifie une demande de travail de flux de travail GitHub Actions avec un nom de flux de travail correspondant à l'expression régulière `\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      FilterGroups:
        - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
        - Type: ACTOR_ACCOUNT_ID
```

```
    Pattern: 12345
    ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
- - Type: HEAD_REF
    Pattern: ^refs/heads/.+
- - Type: FILE_PATH
    Pattern: READ_ME
    ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
- - Type: COMMIT_MESSAGE
    Pattern: \[CodeBuild\]
- - Type: FILE_PATH
    Pattern: ^src/.+|^test/.+
- - Type: EVENT
    Pattern: WORKFLOW_JOB_QUEUED
- - Type: WORKFLOW_NAME
    Pattern: \[CI-CodeBuild\]
```

GitHub webhooks manuels

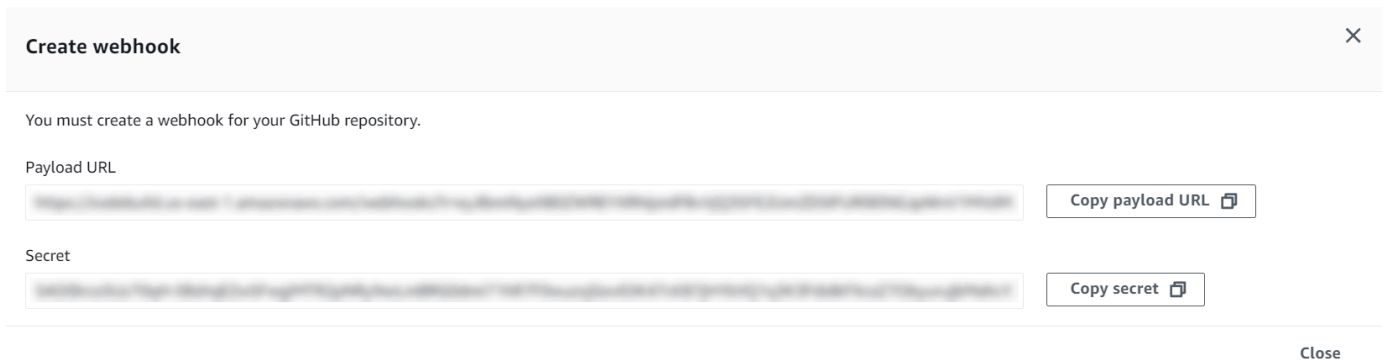
Vous pouvez configurer des GitHub webhooks manuels pour empêcher toute tentative automatique CodeBuild de création d'un webhook à l'intérieur. GitHub CodeBuild renvoie une URL de charge utile dans le cadre de l'appel pour créer le webhook et peut être utilisée pour créer manuellement le webhook à l'intérieur. GitHub Même si vous CodeBuild n'êtes pas autorisé à créer un webhook dans votre GitHub compte, vous pouvez toujours créer manuellement un webhook pour votre projet de construction.

Utilisez la procédure suivante pour créer un webhook GitHub manuel.

Pour créer un GitHub webhook manuel

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Source:
 - Dans le champ Source provider, sélectionnez GitHub.
 - Pour Repository, sélectionnez Repository dans mon GitHub compte.

- Pour Repository URL (URL du référentiel), saisissez **`https://github.com/user-name/repository-name`**
 - Dans les événements Webhook de la source principale :
 - Pour Webhook, facultatif, choisissez Rebuild chaque fois qu'une modification de code est envoyée à ce référentiel.
 - Choisissez Configuration supplémentaire et pour Création manuelle (facultatif), choisissez Créer manuellement un webhook pour ce référentiel dans GitHub la console. .
3. Continuez avec les valeurs par défaut, puis choisissez Create build project. Prenez note de l'URL de charge utile et des valeurs secrètes que vous utiliserez ultérieurement.



Create webhook

You must create a webhook for your GitHub repository.

Payload URL

https://github.com/user-name/repository-name

Copy payload URL

Secret

secret

Copy secret

Close

4. Ouvrez la GitHub console à `https://github.com/user-name/repository-name/settings/hooks` et choisissez Ajouter un webhook.
- Pour URL de charge utile, entrez la valeur de l'URL de charge utile dont vous avez pris note précédemment.
 - Pour le type de contenu, choisissez application/json.
 - Pour Secret, entrez la valeur Secret dont vous avez pris note plus tôt.
 - Configurez les événements individuels auxquels une charge utile de webhook sera envoyée. CodeBuild Pour quels événements souhaitez-vous déclencher ce webhook ? , choisissez Let me select individual events, puis choisissez parmi les événements suivants : Pushes, Pull requests et Releases. Si vous souhaitez démarrer des builds pour des WORKFLOW_JOB_QUEUED événements, choisissez Workflow jobs. Pour en savoir plus sur GitHub les coureurs Actions, consultez [Tutoriel : Configuration d'un CodeBuild exécuteur d'GitHub actions hébergé](#). Pour en savoir plus sur les types d'événements pris en charge par CodeBuild, consultez [GitHub événements webhook](#).
5. Choisissez Ajouter un webhook.

GitHub événements webhook

Vous pouvez utiliser des groupes de filtres Webhook pour spécifier quels événements GitHub Webhook déclenchent une compilation. Par exemple, vous pouvez spécifier qu'une construction n'est déclenchée que pour les modifications apportées à des branches spécifiques.

Vous pouvez créer un ou plusieurs des groupes de filtres webhook pour spécifier les événements webhook qui déclenchent une génération. Une génération est déclenchée si un groupe de filtres est évalué comme vrai, ce qui se produit lorsque tous les filtres du groupe sont évalués comme vrais. Lorsque vous créez un groupe de filtres, vous spécifiez :

Un événement

En GitHub effet, vous pouvez choisir un ou plusieurs des événements suivants : `PUSHPULL_REQUEST_CREATED`, `PULL_REQUEST_UPDATED`, `PULL_REQUEST_REOPENED`, `PULL_REQUEST_MERGED`, `PULL_REQUEST_CLOSED`, `RELEASED`, `PRERELEASED` et `WORKFLOW_JOB_QUEUED`. Le type d'événement du webhook est situé dans l'en-tête `X-GitHub-Event` de la charge utile du webhook. Dans l'en-tête `X-GitHub-Event`, vous pourriez voir `pull_request` ou `push`. Pour un événement de demande d'extraction, le type se situe dans le champ `action` de la charge utile d'événement du webhook. Le tableau suivant présente le mappage des valeurs de l'en-tête `X-GitHub-Event` et des valeurs du champ `action` de la charge utile de la demande d'extraction du webhook aux types d'événements disponibles.

Valeur d'en-tête X-GitHub-Event	Valeur action de la charge utile de l'événement du webhook	Type d'événement
<code>pull_request</code>	<code>opened</code>	<code>PULL_REQUEST_CREATED</code>
<code>pull_request</code>	<code>reopened</code>	<code>PULL_REQUEST_REOPENED</code>
<code>pull_request</code>	<code>synchronize</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pull_request</code>	<code>closed</code> et le champ <code>merged</code> est <code>true</code>	<code>PULL_REQUEST_MERGED</code>
<code>pull_request</code>	<code>closed</code> et le champ <code>merged</code> est <code>false</code>	<code>PULL_REQUEST_CLOSED</code>

Valeur d'en-tête X-GitHub-Event	Valeur action de la charge utile de l'événement du webhook	Type d'événement
push	N/A	PUSH
release	publié	RELEASED
release	pré-publié	PRERELEASED
workflow_job	queued	WORKFLOW_JOB_QUEUED

Note

Le type d'`PULL_REQUEST_REOPENED` événement ne peut être utilisé qu'avec un GitHub serveur GitHub d'entreprise. Le type `PRERELEASED` d'événement `RELEASED` et ne peut être utilisé GitHub qu'avec. Pour plus d'informations sur `WORKFLOW_JOB_QUEUED`, consultez [Tutoriel : Configuration d'un CodeBuild exécuteur d' GitHub actions hébergé](#).

Un ou plusieurs filtres optionnels

Utilisez une expression régulière pour spécifier un filtre. Pour qu'un événement déclenche une génération, chaque filtre du groupe qui lui est associé doit avoir la valeur `true`.

`ACTOR_ACCOUNT_ID(ACTOR_ID dans la console)`

Un événement webhook déclenche une compilation lorsqu'un identifiant de compte GitHub ou un identifiant de compte GitHub Enterprise Server correspond au modèle d'expression régulière. Cette valeur se situe dans la propriété `id` de l'objet `sender` dans la charge utile du webhook.

`HEAD_REF`

Un événement webhook déclenche une construction lorsque la référence principale correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name` ou `refs/tags/tag-name`). Pour un événement d'extraction, le nom de référence est situé dans la propriété `ref` de la charge utile du webhook. Pour les événements de demande d'extraction, le nom de la branche est situé dans la propriété `ref` de l'objet `head` dans la charge utile du webhook.

BASE_REF

Un événement webhook déclenche une construction lorsque la référence de base correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name`). Un filtre `BASE_REF` ne peut être utilisé qu'avec des événements de demande d'extraction. Le nom de la branche se situe dans la propriété `ref` de l'objet base dans la charge utile du webhook.

FILE_PATH

Un webhook déclenche une compilation lorsque le chemin d'un fichier modifié correspond au modèle d'expressions régulières. Un `FILE_PATH` filtre peut être utilisé avec les événements de requêtes GitHub push et pull et les événements push GitHub d'Enterprise Server. Il ne peut pas être utilisé avec les événements de pull request d' GitHubEnterprise Server.

COMMIT_MESSAGE

Un webhook déclenche une compilation lorsque le message de validation principal correspond au modèle d'expression régulière. Un `COMMIT_MESSAGE` filtre peut être utilisé avec les événements de requêtes GitHub push et pull et les événements push GitHub d'Enterprise Server. Il ne peut pas être utilisé avec les événements de pull request d' GitHubEnterprise Server.

TAG_NAME

Un webhook déclenche une compilation lorsque le nom de balise de la version correspond au modèle d'expression régulière. Un `TAG_NAME` filtre peut être utilisé avec les événements de demande GitHub publiés et prépubliés.

RELEASE_NAME

Un webhook déclenche une compilation lorsque le nom de la version correspond au modèle d'expression régulière. Un `RELEASE_NAME` filtre peut être utilisé avec les événements de demande GitHub publiés et prépubliés.

REPOSITORY_NAME

Un webhook déclenche une compilation lorsque le nom du dépôt correspond au modèle d'expression régulière. Un `REPOSITORY_NAME` filtre ne peut être utilisé qu'avec des webhooks GitHub globaux ou organisationnels.

ORGANIZATION_NAME

Un webhook déclenche une compilation lorsque le nom de l'organisation correspond au modèle d'expression régulière. Un ORGANIZATION_NAME filtre ne peut être utilisé qu'avec des webhooks GitHub globaux.

WORKFLOW_NAME

Un webhook déclenche une compilation lorsque le nom du flux de travail correspond au modèle d'expression régulière. Un WORKFLOW_NAME filtre peut être utilisé avec les événements de demande de travail en file d'attente du flux de travail GitHub Actions.

Note

Vous pouvez trouver la charge utile du webhook dans les paramètres du webhook de votre dépôt. GitHub

Rubriques

- [Filtrer les événements du GitHub webhook \(console\)](#)
- [Filtrer les événements du GitHub webhook \(SDK\)](#)
- [Filtrer les événements du GitHub webhook \(CloudFormation\)](#)

Filtrer les événements du GitHub webhook (console)

Suivez les instructions suivantes pour filtrer les événements GitHub webhook à l'aide du AWS Management Console. Pour plus d'informations sur les événements GitHub Webhook, consultez [GitHub événements webhook](#).

Dans Événements webhook de la source principale, sélectionnez ce qui suit. Cette section n'est disponible que lorsque vous avez choisi Repository dans mon GitHub compte pour le référentiel source.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.

3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API AWS CodeBuild .

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour des demandes d'extraction uniquement :

Filter group 1 Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[Start a build under these conditions - optional](#)

[Don't start a build under these conditions - optional](#)

Dans cet exemple de deux groupes de filtres webhook, une génération est déclenchée lorsque l'un des deux ou les deux prennent la valeur true :

- Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/main$` et des références principales qui correspondent à `^refs/heads/branch1$`.

- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour toutes les demandes à l'exception des événements de balise.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_REOPENED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur spécifié GitHub ou par un utilisateur d' GitHub Enterprise Server dont l'ID de compte correspond à l'expression `actor-account-id` régulière.

 Note

Pour plus d'informations sur la façon de trouver votre identifiant de GitHub compte `https://api.github.com/users/user-name`, voir Où *user-name* est votre nom GitHub d'utilisateur ?

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

Remove

► Don't start a build under these conditions - optional

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour un événement de transmission lorsque le message de validation principal correspond à l'expression régulière `\[CodeBuild\]`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - *optional*HEAD_REF - *optional*BASE_REF - *optional*FILE_PATH - *optional*COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une génération pour les événements de travail du flux de travail GitHub Actions uniquement.

 Note

CodeBuild ne traitera les tâches du flux de travail GitHub Actions que si un webhook possède des groupes de filtres contenant le filtre d'événements `WORKFLOW_JOB_QUEUED`.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

► Start a build under these conditions - *optional*► Don't start a build under these conditions - *optional*

Dans cet exemple, un groupe de filtres Webhook déclenche une génération pour un nom de flux de travail qui correspond à l'expression `CI - CodeBuild` régulière.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - *optional*

Filtrer les événements du GitHub webhook (SDK)

Pour utiliser le AWS CodeBuild SDK afin de filtrer les événements Webhook, utilisez le `filterGroups` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Pour plus d'informations sur les événements GitHub Webhook, consultez [GitHub événements webhook](#).

Pour créer un filtre webhook qui déclenche une génération pour des demandes d'extraction uniquement, insérez les éléments suivants dans la syntaxe de demande :

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```

```
        "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    }  
]  
]
```

Pour créer un filtre webhook qui déclenche une génération pour des branches spécifiées uniquement, utilisez le paramètre `pattern` pour spécifier une expression régulière pour filtrer les noms de branche. Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur `true` :

- Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/main$` et des références principales qui correspondent à `^refs/heads/myBranch$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {
```



```
        "type": "HEAD_REF",
        "pattern": "^refs/heads/myBranch$"
    }
]
]
```

Vous pouvez utiliser le paramètre `excludeMatchedPattern` pour spécifier les événements qui ne déclenchent pas une génération. Dans cet exemple, une génération est déclenchée pour toutes les demandes sauf les événements de balise.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/tags/.*",
      "excludeMatchedPattern": true
    }
  ]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière dans l'argument `pattern` changent. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

```
]
```

Dans cet exemple, le groupe de filtres indique qu'une compilation est déclenchée uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^src/.+|^test/.+"  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur spécifié GitHub ou par un utilisateur GitHub Enterprise Server doté d'un identifiant de compte `actor-account-id`.

Note

Pour plus d'informations sur la façon de trouver votre identifiant de GitHub compte <https://api.github.com/users/user-name>, voir Où *user-name* est votre nom GitHub d'utilisateur ?

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",  
      "pattern": "actor-account-id"  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque le message de validation principal correspond à l'expression régulière de l'argument de schéma. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque le message de validation principal de l'événement de transmission correspond à l'expression régulière `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\[CodeBuild\  
  ]  
]
```

Pour créer un filtre Webhook qui déclenche une génération pour les tâches de flux de travail GitHub Actions uniquement, insérez ce qui suit dans la syntaxe de la demande :

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "WORKFLOW_JOB_QUEUED"  
    }  
  ]  
]
```

Filtrer les événements du GitHub webhook (CloudFormation)

Pour utiliser un CloudFormation modèle afin de filtrer les événements Webhook, utilisez la `FilterGroups` propriété du AWS CodeBuild projet.

Pour plus d'informations sur les événements GitHub Webhook, consultez [GitHub événements webhook](#).

La partie au format YAML suivante d'un modèle CloudFormation crée deux groupes de filtres. Ensemble, ils déclenchent une génération lorsque l'un des deux ou les deux prennent la valeur `true` :

- Le premier groupe de filtres indique que les pull requests sont créées ou mises à jour sur les branches dont les noms de référence Git correspondent à l'expression régulière `^refs/heads/main$` d'un GitHub utilisateur qui n'a pas d'identifiant de compte 12345.
- Le deuxième groupe de filtres spécifie les demandes de transmission créées sur des fichiers dont les noms correspondent à l'expression régulière `README` dans des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\[CodeBuild\]`.
- Le quatrième groupe de filtres spécifie une demande de travail de flux de travail GitHub Actions avec un nom de flux de travail correspondant à l'expression régulière `\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
        - Type: ACTOR_ACCOUNT_ID
          Pattern: 12345
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: HEAD_REF
          Pattern: ^refs/heads/.*
```

- Type: FILE_PATH
Pattern: README
ExcludeMatchedPattern: true
- - Type: EVENT
Pattern: PUSH
- Type: COMMIT_MESSAGE
Pattern: \[CodeBuild\]
- Type: FILE_PATH
Pattern: ^src/.+|^test/.+
- - Type: EVENT
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]

GitLab webhooks de groupe

Vous pouvez utiliser des webhooks de CodeBuild GitLab groupe pour démarrer des builds à partir d'événements de webhooks à partir de n'importe quel référentiel au sein d'un GitLab groupe. Les webhooks de groupe fonctionnent avec tous les types d'événements GitLab webhook existants et peuvent être configurés en ajoutant une configuration de portée lors de la création d'un CodeBuild webhook. Vous pouvez également utiliser des webhooks de groupe pour [configurer des GitLab coureurs auto-hébergés CodeBuild au sein](#) d'un même projet afin de recevoir WORKFLOW_JOB_QUEUED des événements provenant de plusieurs référentiels.

Rubriques

- [Configurer un GitLab webhook de groupe](#)
- [Filtrer les événements de webhook de GitLab groupe \(console\)](#)
- [Filtrer les événements webhook de GitLab groupe \(CloudFormation\)](#)

Configurer un GitLab webhook de groupe

Les étapes de haut niveau pour configurer un GitLab webhook de groupe sont les suivantes. Pour plus d'informations sur les GitLab webhooks de groupe, consultez [GitLab webhooks de groupe](#).

1. Définissez l'emplacement source de votre projet
sur CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION.
2. Dans la configuration de la portée du webhook, définissez la portée sur GITLAB_GROUP.

3. Spécifiez un nom dans le cadre de la configuration du scope du webhook. Pour les webhooks de groupe, il s'agit du nom du groupe.

 Note

Si le type de source du projet est `GITLAB_SELF_MANAGED`, vous devrez également spécifier un domaine dans le cadre de la configuration de la portée du webhook.

4. (Facultatif) Si vous souhaitez uniquement recevoir des événements de webhook pour des référentiels spécifiques au sein de votre organisation ou de votre entreprise, vous pouvez les spécifier `REPOSITORY_NAME` sous forme de filtre lors de la création du webhook.
5. Lorsque vous créez un webhook de groupe, assurez-vous qu'il CodeBuild dispose des autorisations nécessaires pour créer des webhooks au niveau du groupe. GitLab Pour ce faire, vous pouvez CodeBuild OAuth cependant utiliser CodeConnections. Pour de plus amples informations, veuillez consulter [GitLab accéder à CodeBuild](#).

Notez que les webhooks de groupe fonctionnent avec tous les types d'événements GitLab webhook existants.

Filtrer les événements de webhook de GitLab groupe (console)

Lorsque vous créez un GitLab projet via la console, sélectionnez les options suivantes pour créer un webhook de GitLab groupe au sein du projet. Pour plus d'informations sur les GitLab webhooks de groupe, consultez [GitLab webhooks de groupe](#).

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Source:
 - Dans le champ Source provider, choisissez GitLabGitLabSelf Managed.
 - Pour Repository, choisissez un GitLabwebhook scopé.

Le GitLab référentiel sera automatiquement défini sur `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, qui est l'emplacement source requis pour les webhooks de groupe.

Note

Lorsque vous utilisez des webhooks de groupe, assurez-vous qu'ils sont CodeBuild autorisés à créer des webhooks au niveau du groupe. GitLab Si vous utilisez une [OAuthconnexion existante](#), vous devrez peut-être la régénérer pour accorder CodeBuild cette autorisation.

Source[Add source](#)**Source 1 - Primary**

Source provider

GitLab

Credential

☒ **Default source credential**
Use your account's default source credential to apply to all projects

☐ **Custom source credential**
Use a custom source credential to override your account's default settings

✓ Successfully connected through CodeConnections - [open resource](#)

[Manage default source credential](#)

Repository

☐ Repository in my GitLab account☒ **GitLab scoped webhook**

Repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

- Dans les événements Webhook de la source principale :
- Dans Nom du groupe, entrez le nom du groupe.

Si le type de source du projet est `GITLAB_SELF_MANAGED`, vous devez également spécifier un domaine dans le cadre de la configuration du groupe Webhook. Par exemple, si l'URL de votre groupe est `https://domain.com/group/group-name`, le domaine est `https://domain.com`.

Note

Ce nom ne peut pas être modifié une fois le webhook créé. Pour modifier le nom, vous pouvez supprimer et recréer le webhook. Si vous souhaitez supprimer complètement le webhook, vous pouvez également mettre à jour l'emplacement de la source du projet vers un GitLab référentiel.

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#)☒ Rebuild every time a code change is pushed to this repository

Group name

Your GitLab group name.

Build type

☒ Single build

Triggers single build

☐ Batch build

Triggers multiple builds as single execution

► **Additional configuration**

- (Facultatif) Dans les groupes de filtres d'événements Webhook, vous pouvez spécifier [les événements pour lesquels vous souhaitez déclencher une nouvelle génération](#). Vous pouvez également spécifier REPOSITORY_NAME comme filtre de manière à ne déclencher des builds que sur des événements webhook provenant de référentiels spécifiques.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Vous pouvez également définir le type d'événement sur WORKFLOW_JOB_QUEUED pour configurer des GitLab coureurs auto-hébergés. Pour de plus amples informations, veuillez consulter [GitLab Coureurs autogérés en AWS CodeBuild](#).

3. Continuez avec les valeurs par défaut, puis choisissez Create build project.

Filtrer les événements webhook de GitLab groupe (CloudFormation)

Pour utiliser un CloudFormation modèle afin de filtrer les événements webhook de groupe, utilisez la ScopeConfiguration propriété du AWS CodeBuild projet. Pour plus d'informations sur les GitLab webhooks de groupe, consultez [GitLab webhooks de groupe](#).

La partie suivante d'un CloudFormation modèle au format YAML crée quatre groupes de filtres. Ensemble, ils déclenchent une compilation lorsqu'une ou toutes les évaluations sont vraies :

- Le premier groupe de filtres indique que les pull requests sont créées ou mises à jour sur les branches dont les noms de référence Git correspondent à l'expression régulière `^refs/heads/main$` d'un GitLab utilisateur qui n'a pas d'identifiant de compte12345.

- Le deuxième groupe de filtres spécifie les demandes de transmission créées sur des fichiers dont les noms correspondent à l'expression régulière `READ_ME` dans des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\[CodeBuild\]`.
- Le quatrième groupe de filtres spécifie un nom de GitLab CI/CD pipeline job request with a CI/CD pipeline correspondant à l'expression régulière `\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
      - Type: ACTOR_ACCOUNT_ID
        Pattern: 12345
        ExcludeMatchedPattern: true
      - - Type: EVENT
        Pattern: PUSH
      - Type: HEAD_REF
        Pattern: ^refs/heads/.*      - Type: FILE_PATH
```

```
Pattern: READ_ME
ExcludeMatchedPattern: true
- - Type: EVENT
Pattern: PUSH
- Type: COMMIT_MESSAGE
Pattern: \[CodeBuild\]
- Type: FILE_PATH
Pattern: ^src/.+|^test/.+
- - Type: EVENT
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]
```

GitLab webhooks manuels

Vous pouvez configurer des GitLab webhooks manuels pour empêcher toute tentative automatique CodeBuild de création d'un webhook à l'intérieur. GitLab CodeBuild renvoie une URL de charge utile dans le cadre de l'appel pour créer le webhook et peut être utilisée pour créer manuellement le webhook à l'intérieur. GitLab Même si vous CodeBuild n'êtes pas autorisé à créer un webhook dans votre GitLab compte, vous pouvez toujours créer manuellement un webhook pour votre projet de construction.

Utilisez la procédure suivante pour créer un webhook GitLab manuel.

Pour créer un GitLab webhook manuel

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Source:
 - Dans le champ Source provider, sélectionnez GitLab.
 - Pour Repository, sélectionnez Repository dans mon GitLab compte.
 - Pour Repository URL (URL du référentiel), saisissez **`https://gitlab.com/user-name/repository-name`**
 - Dans les événements Webhook de la source principale :
 - Pour Webhook, facultatif, choisissez Rebuild chaque fois qu'une modification de code est envoyée à ce référentiel.

- Choisissez Configuration supplémentaire et pour Création manuelle (facultatif), choisissez Créer manuellement un webhook pour ce référentiel dans GitLab la console. .
3. Continuez avec les valeurs par défaut, puis choisissez Create build project. Prenez note de l'URL de charge utile et des valeurs secrètes que vous utiliserez ultérieurement.
 4. Ouvrez la GitLab console à <https://gitlab.com/user-name/repository-name/-/hooks> et choisissez Ajouter un nouveau webhook.
 - Pour URL, entrez la valeur de l'URL de charge utile dont vous avez pris note précédemment.
 - Pour le jeton secret, entrez la valeur secrète dont vous avez pris note précédemment.
 - Configurez les événements individuels auxquels une charge utile de webhook sera envoyée. CodeBuild Pour Trigger, choisissez parmi les événements suivants : événements Push, événements de demande de fusion, événements Releases et événements Job. Pour en savoir plus sur les types d'événements pris en charge par CodeBuild, consultez [GitLab événements webhook](#).
 5. Choisissez Ajouter un webhook.

GitLab événements webhook

Vous pouvez utiliser des groupes de filtres Webhook pour spécifier quels événements GitLab Webhook déclenchent une compilation. Par exemple, vous pouvez spécifier qu'une construction n'est déclenchée que pour les modifications apportées à des branches spécifiques.

Vous pouvez créer un ou plusieurs des groupes de filtres webhook pour spécifier les événements webhook qui déclenchent une génération. Une génération est déclenchée si un groupe de filtres est évalué comme vrai, ce qui se produit lorsque tous les filtres du groupe sont évalués comme vrais. Lorsque vous créez un groupe de filtres, vous spécifiez :

Un événement

En GitLab effet, vous pouvez choisir un ou plusieurs des événements suivants :

PUSHPULL_REQUEST_CREATED,

PULL_REQUEST_UPDATEDPULL_REQUEST_MERGED,PULL_REQUEST_REOPENED,PULL_REQUEST_CLOSED

etWORKFLOW_JOB_QUEUED.

Le type d'événement du webhook est situé dans son en-tête dans le champ X-GitLab-Event. Le tableau suivant présente le mappage des valeurs d'en-tête X-GitLab-Event aux types d'événements. Pour l'événement Merge Request Hook webhook, la charge utile

`object_attributes.action` contiendra des informations supplémentaires sur le type de demande de fusion.

Valeur d'en-tête X-GitLab-Event	object_attributes.action	Type d'événement
Push Hook	N/A	PUSH
Merge Request Hook	ouvrir	PULL_REQUEST_CREATED
Merge Request Hook	mise à jour	PULL_REQUEST_UPDATED
Merge Request Hook	merge	PULL_REQUEST_MERGED
Merge Request Hook	rouvrir	PULL_REQUEST_REOPENED
Merge Request Hook	close	PULL_REQUEST_CLOSED
Release Hook	créer, mettre à jour	RELEASED
Job Hook	N/A	WORKFLOW_JOB_QUEUED

En `PULL_REQUEST_MERGED` effet, si une pull request est fusionnée avec la stratégie squash et que la branche pull request est fermée, le commit de la pull request d'origine n'existe plus. Dans ce cas, la variable d'environnement `CODEBUILD_WEBHOOK_MERGE_COMMIT` contient l'identifiant du commit de fusion écrasé.

Un ou plusieurs filtres optionnels

Utilisez une expression régulière pour spécifier un filtre. Pour qu'un événement déclenche une génération, chaque filtre du groupe qui lui est associé doit avoir la valeur `true`.

`ACTOR_ACCOUNT_ID(ACTOR_ID dans la console)`

Un événement webhook déclenche un build lorsqu'un identifiant de GitLab compte correspond au modèle d'expression régulière. Cette valeur est disponible dans la propriété `account_id` de l'objet `actor` dans la charge utile de filtre du webhook.

HEAD_REF

Un événement webhook déclenche une construction lorsque la référence principale correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name` et `refs/tags/tag-name`). Un filtre `HEAD_REF` évalue le nom de la référence Git de la branche ou de la balise. Le nom de la branche ou de la balise est situé dans le champ `name` de l'objet `new` dans l'objet `push` de la charge utile du webhook. Pour des événements de demande d'extraction, le nom de la branche est situé dans le champ `name` de l'objet `branch` de l'objet `source` dans la charge utile du webhook.

BASE_REF

Un événement webhook déclenche une construction lorsque la référence de base correspond au modèle d'expression régulière. Un filtre `BASE_REF` fonctionne avec des événements de demande d'extraction uniquement (par exemple, `refs/heads/branch-name`). Un filtre `BASE_REF` évalue le nom de la référence Git de la branche. Le nom de la branche est situé dans le champ `name` de l'objet `branch` dans l'objet `destination` de la charge utile du webhook.

FILE_PATH

Un webhook déclenche une compilation lorsque le chemin d'un fichier modifié correspond au modèle d'expression régulière.

COMMIT_MESSAGE

Un webhook déclenche une compilation lorsque le message de validation principal correspond au modèle d'expression régulière.

WORKFLOW_NAME

Un webhook déclenche une compilation lorsque le nom du flux de travail correspond au modèle d'expression régulière.

Note

Vous pouvez trouver la charge utile du webhook dans les paramètres du webhook de votre dépôt. GitLab

Rubriques

- [Filtrer les événements du GitLab webhook \(console\)](#)
- [Filtrer les événements du GitLab webhook \(SDK\)](#)
- [Filtrer les événements du GitLab webhook \(CloudFormation\)](#)

Filtrer les événements du GitLab webhook (console)

Suivez les instructions ci-dessous pour filtrer AWS Management Console les événements Webhook. Pour plus d'informations sur les événements GitLab Webhook, consultez [GitLab événements webhook](#).

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Add filter group (Ajouter un groupe de filtres) pour ajouter un autre groupe de filtres.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API AWS CodeBuild .

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour des demandes d'extraction uniquement :

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕PULL_REQUEST_UPDATED ✕PULL_REQUEST_MERGED ✕

► **Start a build under these conditions - *optional***

► **Don't start a build under these conditions - *optional***

Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur true :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/branch1!`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/branch1$`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ×PULL_REQUEST_UPDATED ×

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Filter 2

Type

Pattern

[Remove](#)

► **Don't start a build under these conditions - optional**

Filter group 2

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ×

Filter 1

Type

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour toutes les demandes à l'exception des événements de balise.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) [×](#)[PULL_REQUEST_CREATED](#) [×](#)[PULL_REQUEST_UPDATED](#) [×](#)[PULL_REQUEST_MERGED](#) [×](#)

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - *optional*

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsqu'une modification est apportée par un GitLab utilisateur dont l'identifiant de compte ne correspond pas à l'expression `actor-account-id` régulière.

Note

Pour plus d'informations sur la façon de trouver votre identifiant de GitLab compte `https://api.github.com/users/user-name`, voir Où *user-name* est votre nom GitLab d'utilisateur ?

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour un événement de transmission lorsque le message de validation principal correspond à l'expression régulière `\[CodeBuild\]`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) [×](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Filtrer les événements du GitLab webhook (SDK)

Pour utiliser le AWS CodeBuild SDK afin de filtrer les événements Webhook, utilisez le `filterGroups` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Pour plus d'informations sur les événements GitLab Webhook, consultez [GitLab événements webhook](#).

Pour créer un filtre webhook qui déclenche une génération pour des demandes d'extraction uniquement, insérez les éléments suivants dans la syntaxe de demande :

```
"filterGroups": [  
  [  
    {  
      "type": "PUSH",  
      "pattern": "\[CodeBuild]\",  
      "filter": true  
    }  
  ]  
]
```

```
{
  "type": "EVENT",
  "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED"
}
]
```

Pour créer un filtre webhook qui déclenche une génération pour des branches spécifiées uniquement, utilisez le paramètre `pattern` pour spécifier une expression régulière pour filtrer les noms de branche. Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur `true` :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/myBranch$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

```
}  
]  
]
```

Vous pouvez utiliser le paramètre `excludeMatchedPattern` pour spécifier les événements qui ne déclenchent pas une génération. Dans cet exemple, une génération est déclenchée pour toutes les demandes à l'exception des événements de balise.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsqu'une modification est apportée par un GitLab utilisateur possédant un identifiant de compte `actor-account-id`.

Note

Pour plus d'informations sur la façon de trouver votre identifiant de GitLab compte <https://api.github.com/users/user-name>, voir Où *user-name* est votre nom GitLab d'utilisateur ?

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",
```



```
    "pattern": "actor-account-id"
  }
]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière dans l'argument `pattern` changent. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

Dans cet exemple, le groupe de filtres indique qu'une compilation est déclenchée uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque le message de validation principal correspond à l'expression régulière de l'argument de schéma. Dans cet

exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque le message de validation principal de l'événement de transmission correspond à l'expression régulière `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\[CodeBuild\  
  ]  
]
```

Filtrer les événements du GitLab webhook (CloudFormation)

Pour utiliser un CloudFormation modèle afin de filtrer les événements Webhook, utilisez la `FilterGroups` propriété du AWS CodeBuild projet. Pour plus d'informations sur les événements GitLab Webhook, consultez [GitLab événements webhook](#).

La partie au format YAML suivante d'un modèle CloudFormation crée deux groupes de filtres. Ensemble, ils déclenchent une génération lorsque l'un des deux ou les deux prennent la valeur `true` :

- Le premier groupe de filtres indique que les pull requests sont créées ou mises à jour sur les branches dont les noms de référence Git correspondent à l'expression régulière `^refs/heads/main$` d'un GitLab utilisateur qui n'a pas d'identifiant de compte 12345.
- Le deuxième groupe de filtres spécifie des demandes de transmission créées sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\[CodeBuild\]`.
- Le quatrième groupe de filtres spécifie une demande de travail de flux de travail GitHub Actions avec un nom de flux de travail correspondant à l'expression régulière `\[CI-CodeBuild\]`.

```
CodeBuildProject:  
  Type: AWS::CodeBuild::Project
```

```
Properties:
  Name: MyProject
  ServiceRole: service-role
  Artifacts:
    Type: NO_ARTIFACTS
  Environment:
    Type: LINUX_CONTAINER
    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/standard:5.0
  Source:
    Type: GITLAB
    Location: source-location
  Triggers:
    Webhook: true
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
      - Type: ACTOR_ACCOUNT_ID
        Pattern: 12345
        ExcludeMatchedPattern: true
      - - Type: EVENT
        Pattern: PUSH
      - Type: HEAD_REF
        Pattern: ^refs/heads/. *
      - - Type: EVENT
        Pattern: PUSH
      - Type: COMMIT_MESSAGE
        Pattern: \[CodeBuild\]
      - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
      - Type: WORKFLOW_NAME
        Pattern: \[CI-CodeBuild\]
```

Webhooks manuels Buildkite

Actuellement, tous CodeBuild les webhooks Buildkite doivent être créés manuellement.

CodeBuild renvoie une URL de charge utile dans le cadre de l'appel pour créer le webhook, qui peut être utilisée pour créer manuellement le webhook dans Buildkite.

Utilisez la procédure suivante pour créer un webhook manuel Buildkite.

Pour créer un CodeBuild projet avec un webhook

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
3. Dans Configuration du projet, choisissez Runner project.

Dans Runner :

- Pour le fournisseur Runner, choisissez Buildkite.
 - Pour le jeton d'agent Buildkite, choisissez Créer un nouveau jeton d'agent en utilisant la page de création d'un secret. Vous serez invité à créer un nouveau secret dans AWS Secrets Manager avec une valeur de secret égale au jeton d'agent Buildkite que vous avez généré ci-dessus.
 - (Facultatif) Si vous souhaitez utiliser des informations d'identification CodeBuild gérées pour votre tâche, sélectionnez le fournisseur du référentiel source de votre tâche dans les options d'identification source de Buildkite et vérifiez que les informations d'identification sont configurées pour votre compte. Vérifiez également que votre pipeline Buildkite utilise Checkout via HTTPS.
4. • Dans Environment (Environnement) :
 - Choisissez une image d'environnement compatible et calculez. Notez que vous avez la possibilité de remplacer les paramètres d'image et d'instance en utilisant une étiquette dans le code YAML de votre flux de travail GitHub Actions. Pour plus d'informations, consultez [Étape 2 : mettez à jour votre flux de travail GitHub Actions YAML](#).
 - Dans Buildspec:
 - Notez que votre spécification de construction sera ignorée à moins qu'elle ne `buildspec-override:true` soit ajoutée sous forme d'étiquette. Au lieu de cela, il le CodeBuild remplacera pour utiliser des commandes qui configureront le coureur auto-hébergé.
 5. Continuez avec les valeurs par défaut, puis choisissez Create build project.
 6. Enregistrez l'URL de charge utile et les valeurs secrètes dans la fenêtre contextuelle Create Webhook. Suivez les instructions affichées dans la fenêtre contextuelle pour créer un nouveau webhook d'organisation Buildkite.

Approbation des commentaires par pull request

CodeBuild prend en charge les politiques de génération de pull request qui fournissent un contrôle supplémentaire sur les builds déclenchés par des pull requests. Il se peut que vous ne souhaitiez pas créer automatiquement des pull requests provenant d'utilisateurs inconnus tant que leurs modifications ne peuvent pas être examinées. Cette fonctionnalité vous permet de demander à l'un des membres de votre équipe de revoir d'abord le code, puis d'exécuter le pipeline. Ceci est couramment utilisé comme mesure de sécurité lors de la création d'un code soumis par des contributeurs inconnus.

Les politiques de génération de pull request vous permettent de contrôler le moment où CodeBuild déclenche les builds pour les pull requests en fonction des autorisations et du statut d'approbation du contributeur. Cela est particulièrement important pour les référentiels publics ou les référentiels qui acceptent les contributions de collaborateurs externes.

Lorsqu'elle est activée, cette fonctionnalité garantit que les builds ne sont déclenchés pour les pull requests que dans les cas suivants :

- La pull request est créée par un contributeur de confiance.
- Un contributeur de confiance approuve la pull request en publiant un commentaire spécifique.

Comment ça marche

Des contributeurs de confiance

Un contributeur fiable est un utilisateur dont le rôle actuel dans le système de contrôle de source est défini dans la politique basée sur les pull requests en tant que rôles d'approbateur. Lorsqu'un contributeur de confiance crée une pull request, il CodeBuild déclenche automatiquement le build, en conservant le comportement actuel.

Contributeurs non fiables

Un contributeur non fiable est un utilisateur dont le rôle n'est pas défini dans la liste des rôles d'approbateur. Lorsqu'une contribution non fiable crée une pull request :

1. CodeBuild marque l'état de construction comme « Echec » avec le message « Pull request approval required for start a build ».
2. Un contributeur fiable doit examiner les modifications et publier un commentaire /
`codebuild_run(<SHA_OF_THE_LATEST_COMMIT>)` pour déclencher le build. Par exemple,
`/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`.

3. CodeBuild valide les autorisations du commentateur et déclenche le build s'il est approuvé.
4. Les résultats du build sont reportés sur la page de pull request.

Syntaxe d'approbation des commentaires

Les contributeurs fiables peuvent approuver les versions en utilisant les formats de commentaires suivants :

- `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`- Les déclencheurs sont basés sur le commit SHA spécifié.

Configuration

Comportement par défaut

La politique de construction par pull request est activée par défaut pour tous les CodeBuild projets nouvellement créés.

Paramètres d'API

La politique de génération de pull request est configurée à l'aide du `PullRequestBuildPolicy` paramètre dans les actions suivantes :

- `CreateWebhook`
- `UpdateWebhook`

PullRequestBuildPolicystructure

```
{
  "requiresCommentApproval": "string",
  "approverRoles": ["string", ...]
}
```

requiresCommentApproval

Spécifie à quel moment une approbation basée sur des commentaires est requise avant de déclencher un build sur des pull requests. Ce paramètre détermine si les builds s'exécutent automatiquement ou nécessitent une approbation explicite par le biais de commentaires.

Type : chaîne

Valeurs valides :

- **DISABLED**- Les builds se déclenchent automatiquement sans nécessiter l'approbation des commentaires.
- **FORK_PULL_REQUESTS**- Seules les pull requests provenant de référentiels bifurqués nécessitent l'approbation des commentaires (sauf si le contributeur est l'un des rôles d'approbateur).
- **ALL_PULL_REQUESTS**- Toutes les pull requests nécessitent l'approbation des commentaires avant l'exécution des builds (sauf si le contributeur est l'un des rôles d'approbateur). C'est la valeur par défaut.

approverRoles

Liste des rôles de référentiel dotés de privilèges d'approbation pour les versions de pull request lorsque l'approbation des commentaires est requise. Seuls les utilisateurs dotés de ces rôles peuvent fournir des approbations de commentaires valides. Si un contributeur de pull request est l'un de ces rôles, ses builds de pull request se déclencheront automatiquement.

Type : tableau de chaînes

Valeurs valides pour les GitHub projets (les valeurs sont mappées aux GitHub rôles) :

- **GITHUB_ADMIN**- Administrateurs de référentiels
- **GITHUB_MAINTAIN**- Mainteneurs de référentiels
- **GITHUB_WRITE**- Utilisateur disposant d'autorisations d'écriture
- **GITHUB_TRIAGE**- Utilisateur disposant d'autorisations de triage
- **GITHUB_READ**- Utilisateur autorisé à lire
- Par défaut : ["GITHUB_ADMIN", "GITHUB_MAINTAINER", "GITHUB_WRITE"]

Valeurs valides pour les GitLab projets (les valeurs sont mappées aux GitLab rôles) :

- **GITLAB_OWNER**- Propriétaire du référentiel
- **GITLAB_MAINTAINER**- Mainteneur du référentiel
- **GITLAB_DEVELOPER**- Utilisateur disposant d'autorisations de développeur
- **GITLAB_REPORTER**- Utilisateur ayant des autorisations de journaliste
- **GITLAB_PLANNER**- Utilisateur disposant d'autorisations de planification
- **GITLAB_GUEST** - Utilisateur autorisé en tant qu'invité
- Par défaut : ["GITLAB_OWNER", "GITLAB_MAINTAINER", "GITLAB_DEVELOPER"]

Valeurs valides pour les projets Bitbucket (les valeurs sont mappées aux rôles Bitbucket) :

- BITBUCKET_ADMIN - Administrateur du référentiel
- BITBUCKET_WRITE- Utilisateur disposant d'autorisations d'écriture
- BITBUCKET_READ - Utilisateur autorisé à lire
- Par défaut : ["BITBUCKET_ADMIN", "BITBUCKET_WRITE"]

Exemples

Activer l'approbation des commentaires pour toutes les pull requests

Pour utiliser le AWS CodeBuild SDK afin d'activer ou de désactiver la politique Pull Request Build pour un webhook, utilisez le `pullRequestBuildPolicy` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Les utilisateurs ayant les rôles d'administrateur, de maintenance et d'écriture sur Github seront traités comme des contributeurs de confiance.

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "ALL_PULL_REQUESTS",
  "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]
}
```

Activer l'approbation des commentaires uniquement pour les administrateurs et les responsables du référentiel

Les utilisateurs ayant GitHub les rôles Admin, Maintain, seront traités comme des contributeurs de confiance.

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "FORK_PULL_REQUESTS",
  "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAINER"]
}
```

Désactiver l'approbation des commentaires

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "DISABLED"
}
```


AWS CloudFormation

Pour utiliser un AWS CloudFormation modèle afin d'activer ou de désactiver la politique Pull Request Build pour un webhook, utilisez la PullRequestBuildPolicy propriété. La partie suivante d'un AWS CloudFormation modèle au format YAML crée un projet avec un webhook pour lequel la politique de génération de demandes d'extraction est activée pour toutes les pull requests. Rôles de maintenance et d'administration tels que spécifiés en tant qu'approbateurs.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
    PullRequestBuildPolicy:
      RequiresCommentApproval: ALL_PULL_REQUESTS
    ApproverRoles:
      - GITHUB_MAINTAIN
      - GITHUB_ADMIN
```

Configuration de la console

Pour utiliser la console de AWS gestion afin de filtrer les événements du webhook, procédez comme suit :

1. Pour l'approbation des commentaires, sélectionnez Désactivé ou activé pour toutes les pull requests (ALL_PULL_REQUEST) ou uniquement pour les pull requests provenant de forks (FORK_PULL_REQUEST).
2. Pour les rôles d'approbateur, sélectionnez des rôles de référentiel dotés de privilèges d'approbation pour les versions de pull request lorsque l'approbation des commentaires est requise.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API CodeBuild .

▼ **Primary source webhook events** [Info](#)

Webhook - optional [Info](#) [↗](#)

☒ Rebuild every time a code change is pushed to this repository

Build type

☒ **Single build**
Triggers single build

☐ **Batch build**
Triggers multiple builds as single execution

Comment approval

ALL_PULL_REQUESTS ▼

Approver roles

BITBUCKET_WRITE ✕

BITBUCKET_ADMIN ✕

► **Webhook event filter groups**

Add filter group

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Afficher les détails d'un projet de construction dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console AWS CLI, ou AWS SDKs pour afficher les détails d'un projet de construction dans CodeBuild.

Rubriques

- [Affichage des détails d'un projet de génération \(console\)](#)

- [Affichage des détails d'un projet de génération \(AWS CLI\)](#)
- [Affichage des détails d'un projet de génération \(AWS SDKs\)](#)

Affichage des détails d'un projet de génération (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

3. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien du projet de génération.
4. Sur la *project-name* page Créer un projet :, choisissez Détails de la construction.

Affichage des détails d'un projet de génération (AWS CLI)

Exécutez la commande batch-get-projects :

```
aws codebuild batch-get-projects --names names
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- *names*: chaîne obligatoire utilisée pour indiquer un ou plusieurs noms de projets de construction sur lesquels afficher les détails. Pour spécifier plusieurs projets de génération, séparez leurs noms par un espace. Vous pouvez spécifier jusqu'à 100 noms de projet de génération. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2
my-other-demo-project
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie. Des ellipses (. . .) sont utilisées pour représenter les données omises par souci de concision.

```
{
  "projectsNotFound": [
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": codebuild-demo-project,
      ...
    },
    {
      ...
      "name": codebuild-demo-project2",
      ...
    }
  ]
}
```

Dans la sortie précédente, le tableau `projectsNotFound` répertorie les noms de projet de génération qui ont été spécifiés, mais pour lesquels aucune information n'a été trouvée. Le tableau `projects` répertorie les détails pour chaque projet de génération pour lequel des informations ont été trouvées. Des détails de projet de génération ont été omis dans la sortie précédente par souci de concision. Pour plus d'informations, consultez la sortie de [Création d'un projet de génération \(AWS CLI\)](#).

La `batch-get-projects` commande ne prend pas en charge le filtrage pour certaines valeurs de propriétés, mais vous pouvez écrire un script qui énumère les propriétés d'un projet. Par exemple, le script shell Linux suivant énumère les projets de la région actuelle pour le compte courant et imprime l'image utilisée par chaque projet.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.
```

```
imageName=""

function getImageName(){
    local environmentValues=(${1//$'\t'/ })
    imageName=${environmentValues[1]}
}

function processProjectInfo() {
    local projectInfo=$1

    while IFS=$'\t' read -r section value; do
        if [[ "$section" == *"ENVIRONMENT"* ]]; then
            getImageName "$value"
        fi
    done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
    if [[ "$projectName" != *"PROJECTS"* ]]; then
        echo "=====

        # Get the detailed information for the project.
        projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

        processProjectInfo "$projectInfo"

        printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
    fi
done
```

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Affichage des détails d'un projet de génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Afficher les noms des projets de construction dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs afficher la liste des projets de construction dans CodeBuild. AWS CLI

Rubriques

- [Affichage d'une liste de noms de projet de génération \(console\)](#)
- [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#)
- [Affichage d'une liste de noms de projet de génération \(AWS SDKs\)](#)

Affichage d'une liste de noms de projet de génération (console)

Vous pouvez consulter la liste des projets de construction dans une AWS région dans la console. Les informations comprennent le nom, le fournisseur source, le référentiel, le statut de la dernière génération et la description, le cas échéant.

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Affichage d'une liste de noms de projet de génération (AWS CLI)

Exécutez la commande `list-projects` :

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- **sort-by**: chaîne facultative utilisée pour indiquer le critère à utiliser pour répertorier les noms de projets de construction. Les valeurs valides sont les suivantes :
 - **CREATED_TIME** : afficher les noms de projet de génération en fonction de la date à laquelle chaque projet de génération a été créé.
 - **LAST_MODIFIED_TIME** : afficher les noms de projet de génération en fonction du moment où les informations sur chaque projet de génération ont été modifiées pour la dernière fois.
 - **NAME** : afficher les noms de projet de génération en fonction du nom de chaque projet de génération.
- **sort-order**: chaîne facultative utilisée pour indiquer l'ordre dans lequel répertorier les projets de construction, en fonction de **sort-by**. Les valeurs valides sont **ASCENDING** et **DESCENDING**.
- **next-token**: chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}
```

Affichage d'une liste de noms de projet de génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

S'intègre AWS CodeBuild

Une construction représente un ensemble d'actions effectuées AWS CodeBuild pour créer des artefacts de sortie (par exemple, un fichier JAR) sur la base d'un ensemble d'artefacts d'entrée (par exemple, une collection de fichiers de classe Java).

Les règles suivantes s'appliquent lorsque vous exécutez plusieurs builds :

- Les builds s'exécutent simultanément, lorsque possible. Le nombre maximal de versions en cours d'exécution simultanée peut varier. Pour de plus amples informations, veuillez consulter [Quotas pour AWS CodeBuild](#).
- Si le projet de génération possède une limite de génération simultanée définie, les versions renvoient une erreur si le nombre de versions en cours atteint la limite de génération simultanée pour le projet. Pour plus d'informations, voir [Activer la limite de génération simultanée](#).
- Si aucune limite de génération simultanée n'est définie pour le projet de génération, les versions sont mises en file d'attente si le nombre de versions en cours atteint la limite de génération simultanée pour la plate-forme et le type de calcul. Le nombre maximal de builds dans une file d'attente est à cinq fois la limite des builds simultanés. Pour de plus amples informations, veuillez consulter [Quotas pour AWS CodeBuild](#).

Un build dans une file d'attente qui ne démarre pas après le nombre de minutes spécifié dans sa valeur d'expiration est supprimé de la file d'attente. La valeur de délai d'expiration par défaut est de huit heures. Vous pouvez remplacer le délai de mise en file d'attente de génération par une valeur comprise entre cinq minutes et huit heures lorsque vous exécutez votre build. Pour de plus amples informations, veuillez consulter [Exécuter AWS CodeBuild les builds manuellement](#).

Il n'est pas possible de prédire l'ordre dans lequel les builds mis en attente démarrent.

Note

Vous pouvez accéder à l'historique d'une génération pour une durée d'un an.

Vous pouvez effectuer ces tâches lorsque vous gérez des générations :

Rubriques

- [Exécuter AWS CodeBuild les builds manuellement](#)

- [Exécuter des builds sur AWS Lambda ordinateur](#)
- [Exécutez des builds sur des flottes à capacité réservée](#)
- [Exécuter des builds par lots](#)
- [Exécuter des tests parallèles dans des builds par lots](#)
- [Des mises en cache pour améliorer les performances](#)
- [Debug s'intègre AWS CodeBuild](#)
- [Supprimer les builds AWS CodeBuild](#)
- [Réessayer les builds manuellement AWS CodeBuild](#)
- [Retry s'intègre automatiquement AWS CodeBuild](#)
- [Arrêter les intégrations AWS CodeBuild](#)
- [Arrêter les compilations par lots AWS CodeBuild](#)
- [Le déclencheur AWS CodeBuild se construit automatiquement](#)
- [Afficher les détails du build dans AWS CodeBuild](#)
- [Afficher la liste des éléments IDs intégrés AWS CodeBuild](#)
- [Afficher la liste des versions IDs d'un projet de construction dans AWS CodeBuild](#)

Exécuter AWS CodeBuild les builds manuellement

Vous pouvez utiliser la AWS CodeBuild AWS CLI console ou AWS SDKs exécuter une intégration CodeBuild.

Rubriques

- [Exécutez les builds localement avec l' AWS CodeBuild agent](#)
- [Exécution d'une génération \(console\)](#)
- [Exécution d'une génération \(AWS CLI\)](#)
- [Exécuter une compilation par lots \(AWS CLI\)](#)
- [Lancement automatique de l'exécution de générations \(AWS CLI\)](#)
- [Arrêt automatique de l'exécution de générations \(AWS CLI\)](#)
- [Exécution d'une génération \(AWS SDKs\)](#)

Exécutez les builds localement avec l' AWS CodeBuild agent

Vous pouvez utiliser l' AWS CodeBuild agent pour exécuter CodeBuild des builds sur une machine locale. Des agents sont disponibles pour les plateformes x86_64 et ARM.

Vous pouvez également vous abonner pour recevoir des notifications lorsque de nouvelles versions de l'agent sont publiées.

Conditions préalables

Avant de commencer, vous devez effectuer les opérations suivantes :

- Installez Git sur votre machine locale.
- Installez et configurez [Docker](#) sur votre machine locale.

Configurer l'image de construction

Vous devez uniquement configurer l'image de construction lors de la première exécution de l'agent ou lorsque l'image a changé.

Pour configurer l'image de construction

1. [Si vous souhaitez utiliser une image Amazon Linux 2 sélectionnée, vous pouvez l'extraire du référentiel CodeBuild public Amazon ECR à l'adresse https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard](https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard) avec la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0
```

Sinon, si vous souhaitez utiliser une autre image Linux, effectuez les opérations suivantes :

- a. Clonez le dépôt CodeBuild d'images :

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Accédez au répertoire des images. Pour cet exemple, utilisez l'aws/codebuild/standard:5.0image :

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Construisez l'image. Cela peut prendre plusieurs minutes.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Téléchargez l' CodeBuild agent.

Pour télécharger la version x86_64 de l'agent, exécutez la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Pour télécharger la version ARM de l'agent, exécutez la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

3. L' CodeBuild agent est disponible à l'[adresse https://gallery.ecr.aws/codebuild/local-builds](https://gallery.ecr.aws/codebuild/local-builds).

La signature SHA (Secure Hash Algorithm) pour la version x86_64 de l'agent est la suivante :

```
sha256:ccb19bdd7af94e4dc761e4c58c267e9455c28ec68d938086b4dc1cf8fe6b0940
```

La signature SHA pour la version ARM de l'agent est la suivante :

```
sha256:7d7b5d35d2ac4e062ae7ba8c662ffed15229a52d09bd0d664a7816c439679192
```

Vous pouvez utiliser le SHA pour identifier la version de l'agent. Pour voir la signature SHA de l'agent, exécutez la commande suivante et recherchez le SHA sous RepoDigests :

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

Exécutez l' CodeBuild agent

Pour exécuter l' CodeBuild agent

1. Accédez au répertoire qui contient la source de votre projet de compilation.
2. Téléchargez le script [codebuild_build.sh](https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh) :

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh
$ chmod +x codebuild_build.sh
```

3. Exécutez le `codebuild_build.sh` script et spécifiez votre image de conteneur et le répertoire de sortie.

Pour exécuter une version x86_64, exécutez la commande suivante :

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Pour exécuter une version ARM, exécutez la commande suivante :

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

Remplacez *<container-image>* par le nom de l'image du conteneur, tel que `aws/codebuild/standard:5.0` ou `public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0`.

Le script lance l'image de construction et exécute la construction sur le projet dans le répertoire en cours. Pour spécifier l'emplacement du projet de génération, ajoutez l'-s *<build project directory>* option à la commande de script.

Recevez des notifications pour les nouvelles versions de CodeBuild l'agent

Vous pouvez vous abonner aux notifications Amazon SNS afin d'être averti lorsque de nouvelles versions de l' AWS CodeBuild agent sont publiées.

Pour vous abonner aux notifications des CodeBuild agents

1. [Ouvrez la console Amazon SNS à l'adresse v3/home. https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. Dans la barre de navigation, si ce n'est pas déjà fait, remplacez la AWS région par USA Est (Virginie du Nord). Vous devez sélectionner cette AWS région car les notifications Amazon SNS auxquelles vous vous abonnez sont créées dans cette région.
3. Dans le panneau de navigation, sélectionnez Abonnements.
4. Sélectionnez Créer un abonnement.
5. Dans Créer un abonnement, procédez comme suit :
 - a. Pour ARN de la rubrique, utilisez l'Amazon Resource Name (ARN) suivant :

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. Pour Protocol (Protocole), choisissez Email (E-mail) ou SMS.
- c. Pour Endpoint (Point de terminaison), choisissez où recevoir les notifications (e-mail ou SMS). Saisissez une adresse e-mail, une adresse ou un numéro de téléphone, y compris l'indicatif régional.
- d. Choisissez Créer un abonnement.
- e. Choisissez E-mail pour recevoir un e-mail vous demandant de confirmer votre inscription. Suivez les instructions dans l'e-mail pour finaliser votre abonnement.

Si vous ne souhaitez plus recevoir ces notifications, exécutez la procédure suivante pour annuler votre abonnement.

Pour vous désabonner des notifications des CodeBuild agents

1. [Ouvrez la console Amazon SNS à l'adresse v3/home. https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. Dans le panneau de navigation, sélectionnez Abonnements.
3. Sélectionnez l'abonnement, puis sous Actions, choisissez Delete subscriptions (Supprimer des abonnements). Lorsque vous êtes invité à confirmer, choisissez Delete.

Exécution d'une génération (console)

À utiliser AWS CodePipeline pour exécuter une compilation avec CodeBuild, ignorez ces étapes et suivez les instructions indiquées dans [À utiliser CodeBuild avec CodePipeline](#).

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de construction, choisissez le projet de construction.
4. Vous pouvez exécuter le build avec les paramètres de projet de build par défaut, ou remplacer les paramètres de build pour ce build uniquement.
 - a. Si vous souhaitez exécuter le build avec les paramètres de projet de build par défaut, choisissez Start build. La construction démarre immédiatement.

- b. Si vous souhaitez remplacer les paramètres par défaut du projet de construction, choisissez Commencer la construction avec des remplacements. Sur la page Démarrer la création, vous pouvez remplacer les éléments suivants :

- Configuration de construction
- Source
- Dérogations de variables d'environnement

Si vous devez sélectionner des remplacements plus avancés, choisissez Advanced build overrides. Dans cette page, vous pouvez annuler les éléments suivants :

- Configuration de construction
- Source
- Environnement
- Spécifications de construction
- Artefacts
- Journaux

Lorsque vous avez effectué vos sélections de remplacement, choisissez Start build.

Pour de plus amples informations sur cette génération, consultez [Affichage des détails d'une génération \(console\)](#).

Exécution d'une génération (AWS CLI)

Note

À utiliser CodePipeline pour exécuter une compilation avec AWS CodeBuild, ignorez ces étapes et suivez les instructions indiquées dans [Créez un pipeline qui utilise CodeBuild \(AWS CLI\)](#).

Pour plus d'informations sur l'utilisation du AWS CLI with CodeBuild, consultez le [Référence des commandes en ligne](#).

1. Exécutez la commande `start-build` de l'une des manières suivantes :

```
aws codebuild start-build --project-name <project-name>
```

Utilisez cette commande si vous souhaitez exécuter une génération utilisant la dernière version de l'artefact d'entrée de génération et les paramètres existants du projet de génération.

```
aws codebuild start-build --generate-cli-skeleton
```

Utilisez cette commande si vous souhaitez exécuter une génération avec une version précédente de l'artefact d'entrée de génération, ou si vous souhaitez remplacer les paramètres pour les artefacts de sortie, les variables d'environnement, le fichier buildspec ou le délai d'expiration de génération par défaut.

2. Si vous exécutez la start-build commande avec l'--project-nameoption, <project-name> remplacez-la par le nom du projet de génération, puis passez à l'étape 6 de cette procédure. Pour obtenir une liste de projets de génération, consultez [Afficher les noms des projets de construction](#).
3. Si vous exécutez la start-build commande avec l'--idempotency-tokenoption, un identifiant ou un jeton unique distinguant majuscules et minuscules est inclus dans la start-build demande. Le jeton est valide pendant 5 minutes après la demande . Si vous répétez la start-build demande avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance des paramètres.
4. Si vous exécutez la commande start-build avec l'option --generate-cli-skeleton, des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, <start-build.json>) à un emplacement de l'ordinateur local ou de l'instance sur laquelle le AWS CLI est installé. Modifiez les données copiées pour qu'elles correspondent au format suivant, et enregistrez vos résultats :

```
{
  "projectName": "<projectName>",
  "sourceVersion": "<sourceVersion>",
  "artifactsOverride": {
    "type": "<type>",
    "location": "<location>",
    "path": "<path>",
    "namespaceType": "<namespaceType>",
    "name": "<artifactsOverride-name>",
    "packaging": "<packaging>"
  },
}
```



```

"buildspecOverride": "buildspecOverride",
"cacheOverride": {
  "location": "cacheOverride-location",
  "type": "cacheOverride-type"
},
"certificateOverride": "certificateOverride",
"computeTypeOverride": "computeTypeOverride",
"environmentTypeOverride": "environmentTypeOverride",
"environmentVariablesOverride": {
  "name": "environmentVariablesOverride-name",
  "value": "environmentVariablesValue",
  "type": "environmentVariablesOverride-type"
},
"gitCloneDepthOverride": "gitCloneDepthOverride",
"imageOverride": "imageOverride",
"idempotencyToken": "idempotencyToken",
"insecureSslOverride": "insecureSslOverride",
"privilegedModeOverride": "privilegedModeOverride",
"queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
"reportBuildStatusOverride": "reportBuildStatusOverride",
"timeoutInMinutesOverride": "timeoutInMinutesOverride",
"sourceAuthOverride": "sourceAuthOverride",
"sourceLocationOverride": "sourceLocationOverride",
"serviceRoleOverride": "serviceRoleOverride",
"sourceTypeOverride": "sourceTypeOverride"
}

```

Remplacez les espaces réservés suivants :

- **projectName**: chaîne obligatoire. Nom du projet de génération à utiliser pour la génération.
- **sourceVersion**: chaîne facultative. Une version du code source à générer, comme suit :
 - Pour Amazon S3, l'ID de version qui correspond à la version du fichier ZIP d'entrée que vous souhaitez créer. Si **sourceVersion** ce n'est pas spécifié, c'est la dernière version qui est utilisée.
 - Pour CodeCommit, l'ID de validation qui correspond à la version du code source que vous souhaitez créer. S'il n'**sourceVersion**est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé. (Vous ne pouvez pas spécifier de nom de balise pour **sourceVersion**, mais vous pouvez spécifier l'ID de validation de la balise.)
 - Pour GitHub, l'ID de validation, l'ID de pull request, le nom de branche ou le nom de balise correspondant à la version du code source que vous souhaitez créer. Si un ID de demande

d'extraction est spécifié, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'*sourceVersion* est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.

- Pour Bitbucket, l'ID de validation, le nom de branche ou le nom de balise qui correspond à la version du code source à générer. Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'*sourceVersion* est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Les espaces réservés suivants sont pour `artifactsOverride`.
 - *type* - Facultatif. Type d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *location* - Facultatif. Emplacement d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *path* - Facultatif. Chemin d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *namespaceType* - Facultatif. Type de chemin d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *name* - Facultatif. Nom d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *packaging* - Facultatif. Type de packaging d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
- *buildspecOverride* - Facultatif. Déclaration de spécification de génération (buildspec) qui remplace pour cette génération celle définie dans le projet de génération. Si cette valeur est définie, il peut s'agir d'une définition de spécification de génération en ligne, du chemin vers un fichier buildspec alternatif, relatif à la valeur de la variable d'environnement `CODEBUILD_SRC_DIR` intégrée, ou du chemin vers un compartiment S3. Le compartiment S3 doit être dans la même région AWS que le projet de génération. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Si cette valeur n'est pas fournie ou qu'elle est définie sur une chaîne vide, le code source doit contenir un fichier `buildspec.yml` dans son répertoire racine. Pour de plus amples informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).
- Les espaces réservés suivants sont pour `cacheOverride`.

- *cacheOverride-location* - Facultatif. Emplacement d'un objet ProjectCache pour cette génération qui remplace l'objet ProjectCache spécifié dans le projet de génération. *cacheOverride* est facultatif et accepte un objet ProjectCache. *location* est obligatoire dans un objet ProjectCache.
- *cacheOverride-type* - Facultatif. Type d'un objet ProjectCache pour cette génération qui remplace l'objet ProjectCache spécifié dans le projet de génération. *cacheOverride* est facultatif et accepte un objet ProjectCache. *type* est obligatoire dans un objet ProjectCache.
- *certificateOverride* - Facultatif. Nom d'un certificat pour cette génération qui remplace celui spécifié dans le projet de génération.
- *environmentTypeOverride* - Facultatif. Type de conteneur pour cette génération qui remplace celui spécifié dans le projet de génération. La chaîne valide actuelle est LINUX_CONTAINER.
- Les espaces réservés suivants sont pour *environmentVariablesOverride*.
 - *environmentVariablesOverride-name* - Facultatif. Nom d'une variable d'environnement dans le projet de génération dont vous voulez substituer la valeur pour cette génération.
 - *environmentVariablesOverride-type* - Facultatif. Type de variable d'environnement dans le projet de génération dont vous voulez substituer la valeur pour cette génération.
 - *environmentVariablesValue* - Facultatif. Valeur de la variable d'environnement définie dans le projet de génération que vous voulez substituer pour cette génération.
- *gitCloneDepthOverride* - Facultatif. Valeur de Git clone depth dans le projet de génération dont vous voulez remplacer la valeur pour cette génération. Si votre type de source est Amazon S3, cette valeur n'est pas prise en charge.
- *imageOverride* - Facultatif. Nom d'une image pour cette génération qui remplace celle spécifiée dans le projet de génération.
- *idempotencyToken* - Facultatif. Chaîne qui tient lieu de jeton pour spécifier que la demande de génération est idempotente. Vous pouvez choisir n'importe quelle chaîne de 64 caractères maximum. Le jeton est valide pendant 5 minutes après la demande de démarrage de la construction. Si vous répétez la demande de démarrage de la construction avec le même jeton, mais que vous modifiez un paramètre, cela CodeBuild renvoie une erreur de non-concordance des paramètres.
- *insecureSslOverride*: booléen facultatif qui indique s'il faut remplacer le paramètre TLS non sécurisé spécifié dans le projet de génération. Le paramètre TLS non sécurisé détermine

s'il convient d'ignorer les avertissements TLS lors de la connexion au code source du projet. Cette dérogation s'applique uniquement si la source du build est GitHub Enterprise Server.

- ***privilegedModeOverride***: booléen facultatif. Si celle-ci est définie sur true, la génération remplace le mode privilégié dans le projet de génération.
- ***queuedTimeoutInMinutesOverride***: entier facultatif qui indique le nombre de minutes pendant lesquelles une version est autorisée à être mise en file d'attente avant son expiration. Sa valeur minimale est de cinq minutes et sa valeur maximale est 480 minutes (huit heures).
- ***reportBuildStatusOverride***: booléen facultatif qui indique s'il faut envoyer à votre fournisseur source l'état du début et de la fin d'une compilation. Si vous le définissez avec un fournisseur de source autre qu' GitHub GitHub Enterprise Server ou Bitbucket, un `invalidInputException` est émis.
- ***sourceAuthOverride***: chaîne facultative. Type d'autorisation pour cette génération qui remplace celle définie dans le projet de génération. Cette dérogation s'applique uniquement si la source du projet de build est Bitbucket ou. GitHub
- ***sourceLocationOverride***: chaîne facultative. Emplacement qui remplace l'emplacement de la source pour cette génération par celui défini dans le projet de génération.
- ***serviceRoleOverride***: chaîne facultative. Nom d'un rôle de service pour cette génération qui remplace celui spécifié dans le projet de génération.
- ***sourceTypeOverride***: chaîne facultative. Type d'entrée source pour cette génération qui remplace l'entrée source définie dans le projet de génération. Les chaînes valides sont NO_SOURCE, CODECOMMIT, CODEPIPELINE, GITHUB, S3, BITBUCKET et GITHUB_ENTERPRISE.
- ***timeoutInMinutesOverride***: Numéro optionnel. Nombre de minutes de délai d'expiration de génération qui remplace pour cette de génération celui défini dans le projet de génération.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un identifiant de clé d' AWS accès, une clé d'accès AWS secrète ou un mot de passe en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store. CodeBuild peut utiliser un paramètre stocké dans Amazon EC2 Systems Manager Parameter Store uniquement si le nom de ce paramètre commence par /CodeBuild/ (par exemple,/CodeBuild/dockerLoginPassword). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Sélectionnez Créer un paramètre, puis suivez les instructions. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez éventuellement spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager

utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour le déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence le paramètre `/CodeBuild/` tel qu'il est stocké. Toutefois, si vous utilisez la console Amazon EC2 Systems Manager Parameter Store pour créer un paramètre, vous devez commencer le nom du paramètre par et définir Type sur Secure String. `/CodeBuild/` Pour plus d'informations, consultez les [AWS Systems Manager sections Parameter Store](#) et [Walkthrough : Create and test a String parameter \(console\)](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters`. Si vous avez choisi Créer un nouveau rôle de service dans votre compte plus tôt, cette action est automatiquement incluse dans le rôle de service par défaut de votre projet de génération. CodeBuild Toutefois, si vous avez choisi précédemment Choisir un rôle de service existant dans votre compte, vous devez inclure séparément cette action dans votre rôle de service.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur de la variable d'environnement est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration du fichier `buildspec` a la priorité la plus faible.

Pour de plus amples informations sur les valeurs valides pour ces espaces réservés, consultez [Création d'un projet de génération \(AWS CLI\)](#). Pour obtenir une liste des derniers paramètres pour un projet de génération, consultez [Afficher les détails du projet de construction](#).

5. Placez-vous dans le répertoire contenant le fichier que vous venez d'enregistrer, puis exécutez à nouveau la commande `start-build`.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. En cas de réussite, des données similaires à ce qui est décrit dans la procédure [Pour exécuter la génération](#) apparaissent en sortie.

Pour utiliser les informations détaillées sur cette génération, notez la valeur `id` de la sortie, puis consultez [Affichage des détails d'une génération \(AWS CLI\)](#).

Exécuter une compilation par lots (AWS CLI)

1. Exécutez la commande `start-build-batch` de l'une des manières suivantes :

```
aws codebuild start-build-batch --project-name <project-name>
```

Utilisez cette commande si vous souhaitez exécuter une génération utilisant la dernière version de l'artefact d'entrée de génération et les paramètres existants du projet de génération.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Utilisez cette commande si vous souhaitez exécuter une génération avec une version précédente de l'artefact d'entrée de génération, ou si vous souhaitez remplacer les paramètres pour les artefacts de sortie, les variables d'environnement, le fichier `buildspec` ou le délai d'expiration de génération par défaut.

2. Si vous exécutez la `start-build-batch` commande avec l'option `--project-name`, `<project-name>` remplacez-la par le nom du projet de génération, puis passez à l'étape 6 de cette procédure. Pour obtenir une liste de projets de génération, consultez [Afficher les noms des projets de construction](#).
3. Si vous exécutez la `start-build-batch` commande avec l'option `--idempotency-token`, un identifiant unique distinguant majuscules et minuscules, ou jeton, est inclus dans la `start-`

build-batch demande. Le jeton est valide pendant 5 minutes après la demande . Si vous répétez la start-build-batch demande avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance des paramètres.

4. Si vous exécutez la start-build-batch commande avec l'option `--generate-cli-skeleton`, les données au format JSON sont sorties dans le fichier. *<json-file>* Ce fichier est similaire au squelette produit par la start-build commande, avec l'ajout de l'objet suivant. Pour plus d'informations sur les objets courants, consultez [Exécution d'une génération \(AWS CLI\)](#).

Modifiez ce fichier pour ajouter des remplacements de build et enregistrez vos résultats.

```
"buildBatchConfigOverride": {
  "combineArtifacts": combineArtifacts,
  "restrictions": {
    "computeTypesAllowed": [
      allowedComputeTypes
    ],
    "maximumBuildsAllowed": maximumBuildsAllowed
  },
  "serviceRole": "batchServiceRole",
  "timeoutInMins": batchTimeout
}
```

L'buildBatchConfigOverrideobjet est une [ProjectBuildBatchConfig](#)structure qui contient les remplacements de configuration de génération par lots pour cette version.

combineArtifacts

Un booléen qui indique si les artefacts de construction pour la génération par lots doivent être combinés dans un seul emplacement d'artefact.

allowedComputeTypes

Tableau de chaînes qui spécifient les types de calcul autorisés pour la construction par lots. Reportez-vous [à la section Types de calcul de l'environnement](#) de génération pour ces valeurs.

maximumBuildsAllowed

Spécifie le nombre maximal de builds autorisés.

batchServiceRole

Spécifie l'ARN du rôle de service pour le projet de génération par lots.

batchTimeout

Spécifie la durée maximale, en minutes, pendant laquelle la génération du lot doit être terminée.

5. Placez-vous dans le répertoire contenant le fichier que vous venez d'enregistrer, puis exécutez à nouveau la commande `start-build-batch`.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. En cas de succès, la représentation JSON d'un [BuildBatch](#) objet apparaît dans la sortie de la console. Voir la [syntaxe de StartBuildBatch réponse](#) pour un exemple de ces données.

Lancement automatique de l'exécution de générations (AWS CLI)

Si votre code source est stocké dans un référentiel GitHub ou dans un référentiel GitHub Enterprise Server, vous pouvez utiliser des GitHub webhooks pour AWS CodeBuild reconstruire votre code source chaque fois qu'une modification de code est envoyée au référentiel.

Exécutez la commande `create-webhook` comme suit :

```
aws codebuild create-webhook --project-name <project-name>
```

<project-name> est le nom du projet de construction qui contient le code source à reconstruire.

En GitHub effet, des informations similaires aux suivantes apparaissent dans la sortie :

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

<url> est l'URL du GitHub webhook.

Pour GitHub Enterprise Server, des informations similaires aux suivantes apparaissent dans la sortie :


```
{
  "webhook": {
    "secret": "YRV4YJAGfsekJiirp5ytx86oZpyhUdySNSDTLNUxOXX1c7aZ6XYDf37-ZFYy02rs4JSE70mLW3w-gh-ryvOB80SSSC1aAtBtuPKhW
YuncCCmdogCVCfni07ukYX2_xM--n1Dma5EngIe_Bi_N465yi33zyTUNPo1qxCPLO-BwghcVa9lAurwR77-uy7i-_XC3FahwMx1f4ubogBB5mMT2A16apjq
QJOakSb61XVKYy21Giuy4nliAXfv9Wnn76CaCsndb3fVIE78fpygfo41xyS5Q6vpo6LRTKtPzbyeThbVXGda1P3vnkBlmKMD08RTgI1m2oYr17dwiZiQlrvo
ColNgy1S00_7LKfA-nNXFc_f1SiFy0AqeMB43-d00cdkzybHncE81QTRwEUCfmx-AJCwmLVX0kg06G7T92Sjbpz0fR1kh5pwIF193_bb_jOHdInK6iOiPpf2
dIDAIZGhMaggzveghe-axDeTAboPu08J6gfI1yK05aq9q15d3z1C1PERusM6JfTJR_a-z-L_kylr-4h5Sxas5JNj43_X0BRwqT51xqvH-A69bV07kbvt_Kc6wxk
WYsYCEMoa_PFa72QgyfY6GB00gMNj31fyFbjthORNL1Cd06-3J-McDLoYrTSEOV9QnxvsG5zu1Ns-z2brk3Jtg_MOFNwocfUxtFxb7vrGdUHLr1dZLxLrK
xOVVuDUMw9vHwMr-hUkeGo_1kDKyK4E20FvZxPjYw0vFfV-dwxFRR_mlfzKx1Wyfmt21fTlKp_Yzj_4WeFAckGefr-ilNayvsZpzj78Ae1adVolF48AmDn
2pW5WjJatU9zt942GelisFFmKakcvJuy5yxXhaxxbhUdyC8NHYiESUWPfcfngrMsr0p3P4AUChIpiZCYyuiwI_cac-pIUBO8Xaur_lu_fyFghg0Jc7cfTnA36
rsvXS5ndFDM8p3HNBEljaF9Q26AijgePwTHIKJON3AduWpKz_hWtXyUoA8MDZfPXT3bBoT6N5Z5THBHSYXR",
    "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?2=eyJ1bmMleYXB0ZWREYXRhIjoiaUmfQmMjERGRQbGhwL2NT
Nld3R0VG9jZ20TnwLz1ZVG1Nz1pRIER0RUsxdzhGehWnVFQwTR0WEvTwdJRnNmRkZ3S3Rnc0xYMEncXFTagk1e1nSy9zPSi5Im1dUGFyYW1ldGvyU3B1
YyI6IndsQ1Qrc2VPQjBCZzhPeVYlLCJtYXR1cmhhbFNLdFNlcmhhbC1mX0%3D&v=1"
  }
}
```

1. Copiez la clé secrète et l'URL de la charge utile à partir de la sortie. Vous en avez besoin pour ajouter un webhook dans GitHub Enterprise Server.
2. Dans GitHub Enterprise Server, choisissez le référentiel dans lequel votre CodeBuild projet est stocké. Choisissez Settings (Paramètres), Hooks & services, puis Add webhook (Ajouter un webhook).
3. Entrez l'URL et la clé secrète de la charge utile, acceptez les valeurs par défaut pour les autres champs, puis choisissez Add webhook.

Arrêt automatique de l'exécution de générations (AWS CLI)

Si votre code source est stocké dans un référentiel GitHub ou dans un référentiel GitHub Enterprise Server, vous pouvez configurer des GitHub webhooks pour AWS CodeBuild reconstruire votre code source chaque fois qu'une modification de code est envoyée au référentiel. Pour de plus amples informations, veuillez consulter [Lancement automatique de l'exécution de générations \(AWS CLI\)](#).

Si vous avez activé ce comportement, vous pouvez le désactiver en exécutant la commande `delete-webhook` comme suit :

```
aws codebuild delete-webhook --project-name <project-name>
```

- où *<project-name>* est le nom du projet de construction qui contient le code source à reconstruire.

Si cette commande aboutit, aucune information ni aucune erreur n'apparaît dans la sortie.

 Note

Cela supprime le webhook de votre CodeBuild projet uniquement. Vous devez également supprimer le webhook de votre référentiel GitHub ou de celui GitHub d'Enterprise Server.

Exécution d'une génération (AWS SDKs)

À utiliser CodePipeline pour exécuter une compilation avec AWS CodeBuild, ignorez ces étapes et suivez [Utilisez AWS CodeBuild with AWS CodePipeline pour tester le code et exécuter des builds](#) plutôt les instructions indiquées.

Pour plus d'informations sur l'utilisation CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Exécuter des builds sur AWS Lambda ordinateur

AWS Lambda compute offre des vitesses de démarrage optimisées pour vos builds. AWS Lambda prend en charge des builds plus rapides grâce à une latence de démarrage plus faible. AWS Lambda évolue également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Cependant, certains cas d'utilisation AWS Lambda ne sont pas compatibles, et s'ils vous concernent, utilisez le calcul EC2. Pour de plus amples informations, veuillez consulter [Limites du AWS Lambda calcul](#).

Rubriques

- [Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda](#)
- [Et si l'image sélectionnée n'inclut pas les outils dont j'ai besoin ?](#)
- [Dans quelles régions le AWS Lambda calcul CodeBuild est-il possible ?](#)
- [Limites du AWS Lambda calcul](#)
- [Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java](#)
- [Créez une application React d'une seule page avec CodeBuild Lambda Node.js](#)
- [Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python](#)

Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda

AWS Lambda prend en charge les outils suivants : AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby et .NET.

Et si l'image sélectionnée n'inclut pas les outils dont j'ai besoin ?

Si l'image sélectionnée n'inclut pas les outils dont vous avez besoin, vous pouvez fournir une image Docker d'environnement personnalisée qui inclut les outils nécessaires.

Note

Lambda ne prend pas en charge les fonctions qui utilisent les images de conteneur multi-architecture. Pour plus d'informations, voir [Création d'une fonction Lambda à l'aide d'une image de conteneur](#) dans le Guide du AWS Lambda développeur.

Notez que vous avez besoin des autorisations Amazon ECR suivantes pour utiliser des images personnalisées pour le calcul Lambda :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
```

```

        "ecr:BatchGetImage"
    ],
    "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/image-  

repo"
    }
  ]
}

```

Notez également que cela wget doit être installé curl ou doit être installé pour pouvoir utiliser des images personnalisées.

Dans quelles régions le AWS Lambda calcul CodeBuild est-il possible ?

En CodeBuild, le AWS Lambda calcul est pris en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort), Europe (Irlande) et Amérique du Sud (São Paulo). Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Limites du AWS Lambda calcul

Certains cas d'utilisation AWS Lambda ne sont pas compatibles, et s'ils vous concernent, utilisez le calcul EC2 :

- AWS Lambda ne prend pas en charge les outils qui nécessitent des autorisations root. Pour des outils tels que yum ou rpm, utilisez le type de calcul EC2 ou d'autres outils ne nécessitant pas d'autorisations root.
- AWS Lambda ne prend pas en charge les builds ou les exécutions de Docker.
- AWS Lambda ne prend pas en charge l'écriture dans des fichiers externes/tmp. Les gestionnaires de packages inclus sont configurés pour utiliser le /tmp répertoire par défaut pour le téléchargement et le référencement des packages.
- AWS Lambda ne prend pas en charge le type d'environnement LINUX_GPU_CONTAINER et n'est pas pris en charge sur Windows Server Core 2019.
- AWS Lambda ne prend pas en charge la mise en cache, les délais de génération personnalisés, les délais d'attente, les badges de génération, le mode privilégié, les environnements d'exécution personnalisés ou les durées d'exécution supérieures à 15 minutes.

- AWS Lambda ne prend pas en charge la connectivité VPC, une plage fixe d'adresses IP CodeBuild sources, le système EFS, l'installation de certificats ou l'accès SSH avec le gestionnaire de session.

Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java

The AWS Serverless Application Model (AWS SAM) est un framework open source permettant de créer des applications sans serveur. Pour plus d'informations, consultez le [AWS Serverless Application Model référentiel](#) sur GitHub. L'exemple Java suivant utilise Gradle pour créer et tester une AWS Lambda fonction. Ensuite, la AWS SAM CLI est utilisée pour déployer le CloudFormation modèle et le bundle de déploiement. Grâce à CodeBuild Lambda, les étapes de construction, de test et de déploiement sont toutes gérées automatiquement, ce qui permet de mettre à jour rapidement l'infrastructure sans intervention manuelle lors d'une seule génération.

Configurez votre AWS SAM référentiel

Créez un AWS SAM Hello World projet à l'aide de la AWS SAM CLI.

Pour créer votre AWS SAM projet

1. Suivez les instructions du guide du AWS Serverless Application Model développeur pour [installer la AWS SAM CLI](#) sur votre machine locale.
2. Exécutez `sam init` et sélectionnez la configuration de projet suivante.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions? [y/N]: N
Project name [sam-app]: <insert project name>
```

3. Téléchargez le dossier AWS SAM du projet dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Java

Créez un projet AWS CodeBuild Lambda Java et configurez les autorisations IAM nécessaires pour le build.

Pour créer votre projet CodeBuild Lambda Java

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.
5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Java.
 - Pour Image, sélectionnez aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21.
 - Pour Rôle de service, laissez l'option Nouveau rôle de service sélectionnée. Notez le nom du rôle. Cela sera nécessaire lorsque vous mettrez à jour les autorisations IAM du projet ultérieurement dans cet exemple.
6. Choisissez Créer un projet de génération.
7. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
8. Dans le volet de navigation, choisissez Rôles et sélectionnez le rôle de service associé à votre projet. Vous pouvez trouver votre rôle dans le projet en CodeBuild sélectionnant votre projet de construction, en choisissant Modifier, Environnement, puis Rôle de service.
9. Sélectionnez l'onglet Trust relationships (Relations d'approbation), puis Edit trust policy (Modifier la politique d'approbation).

10. Ajoutez la politique intégrée suivante à votre rôle IAM. Cela sera utilisé pour déployer votre AWS SAM infrastructure ultérieurement. Pour plus d'informations, consultez [Ajout et suppression d'autorisations basées sur l'identité IAM](#) dans le Guide de l'utilisateur IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "iam:*",
        "apigateway:*",
        "s3:*"
      ],
      "Resource": "arn:aws:iam::*:role/Service*"
    }
  ]
}
```

Configurer le buildspec du projet

Pour créer, tester et déployer votre fonction Lambda, CodeBuild lit et exécute les commandes de construction à partir d'une spécification de construction.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Passer à l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
env:
  variables:
```

```
GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
      - sam build --template-file template.yaml
      - echo "Running deploy..."
      - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
      - yes | sam deploy
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Déployez votre infrastructure AWS SAM Lambda

Utilisez CodeBuild Lambda pour déployer automatiquement votre infrastructure Lambda

Pour déployer votre infrastructure Lambda

1. Choisissez Démarrer la génération. Cela permettra de créer, de tester et de déployer automatiquement votre AWS SAM application pour AWS Lambda l'utiliser CloudFormation.
2. Une fois le build terminé, accédez à la AWS Lambda console et recherchez votre nouvelle fonction Lambda sous le nom du AWS SAM projet.
3. Testez votre fonction Lambda en sélectionnant API Gateway dans la vue d'ensemble des fonctions, puis en cliquant sur l'URL du point de terminaison de l'API. Vous devriez voir une page s'ouvrir avec le message "message": "hello world".

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées par votre AWS SAM modèle et CodeBuild.

Pour nettoyer votre infrastructure

1. Accédez à la CloudFormation console et sélectionnez leaws-sam-cli-managed-default.
2. Dans Ressources, videz le compartiment de déploiementSamCliSourceBucket.
3. Supprimez la aws-sam-cli-managed-default pile.

4. Supprimez la CloudFormation pile associée à votre AWS SAM projet. Cette pile doit porter le même nom que votre AWS SAM projet.
5. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.
6. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.

Créez une application React d'une seule page avec CodeBuild Lambda Node.js

[Create React App](#) est un moyen de créer des applications React d'une seule page. L'exemple de fichier Node.js suivant utilise Node.js pour créer les artefacts source à partir de Create React App et renvoie les artefacts de construction.

Configuration de votre référentiel source et de votre compartiment d'artefacts

Créez un référentiel source pour votre projet à l'aide de yarn et de Create React App.

Pour configurer le référentiel source et le compartiment d'artefacts

1. Sur votre machine locale, lancez-le `yarn create react-app <app-name>` pour créer une application React simple.
2. Téléchargez le dossier du projet de l'application React dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Node.js

Créez un projet AWS CodeBuild Lambda Node.js.

Pour créer votre projet CodeBuild Lambda Node.js

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.
5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Node.js.
 - Pour Image, sélectionnez aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.
6. Dans Artefacts:
 - Pour Type, sélectionnez Amazon S3.
 - Dans le champ Nom du compartiment, sélectionnez le compartiment d'artefacts du projet que vous avez créé précédemment.
 - Pour l'emballage des artefacts, sélectionnez Zip.
7. Choisissez Créer un projet de génération.

Configurer le buildspec du projet

Pour créer votre application React, CodeBuild lit et exécute les commandes de construction à partir d'un fichier buildspec.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Basculer vers l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
phases:
  build:
    commands:
      - yarn
      - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
      - yarn run build
```

```
- yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
artifacts:
  name: "build-output"
  files:
    - "**/*"
reports:
  test-report:
    files:
      - 'junit.xml'
    file-format: 'JUNITXML'
  coverage-report:
    files:
      - 'coverage/clover.xml'
    file-format: 'CLOVERXML'
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Créez et exécutez votre application React

Créez l'application React sur CodeBuild Lambda, téléchargez les artefacts de construction et exécutez l'application React localement.

Pour créer et exécuter votre application React

1. Choisissez Démarrer la génération.
2. Une fois le build terminé, accédez au bucket d'artefacts de votre projet Amazon S3 et téléchargez l'artefact de l'application React.
3. Décompressez l'artefact de construction de React et placez-le `run npm install -g serve && serve -s build` dans le dossier du projet.
4. La `serve` commande servira le site statique sur un port local et imprimera le résultat sur votre terminal. Vous pouvez visiter l'URL localhost ci-dessous Local : dans la sortie du terminal pour afficher votre application React.

Pour en savoir plus sur la gestion du déploiement d'un serveur basé sur React, voir [Créer un déploiement d'applications React](#).

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées pour votre CodeBuild projet.

Pour nettoyer votre infrastructure

1. Supprimer les artefacts de votre projet (compartiment Amazon S3)
2. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.
3. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.

Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python

L'exemple Python suivant utilise [Boto3](#) et Lambda CodeBuild Python pour mettre à jour la configuration d'une fonction Lambda. Cet exemple peut être étendu pour gérer d'autres AWS ressources par programmation. Pour plus d'informations, consultez la documentation de [Boto3](#).

Conditions préalables

Créez ou trouvez une fonction Lambda dans votre compte.

Cet exemple suppose que vous avez déjà créé une fonction Lambda dans votre compte et que vous l'utiliserez CodeBuild pour mettre à jour les variables d'environnement de la fonction Lambda. Pour plus d'informations sur la configuration d'une fonction Lambda via CodeBuild, consultez l'[Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java](#) exemple ou rendez-vous sur [AWS Lambda](#)

Configurez votre référentiel de sources

Créez un dépôt source pour stocker votre script python Boto3.

Pour configurer le référentiel source

1. Copiez le script python suivant dans un nouveau fichier appelé `update_lambda_environment_variables.py`.

```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
    print("Updating lambda function " + lambda_function_name + " environment
variable "
        + lambda_env_variable + " to " + lambda_env_variable_value)
    lambda_client.update_function_configuration(
        FunctionName=lambda_function_name,
        Environment={
            'Variables': {
                lambda_env_variable: lambda_env_variable_value
            }
        },
    )

if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Téléchargez le fichier python dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Python

Créez un projet CodeBuild Lambda Python.

Pour créer votre projet CodeBuild Lambda Java

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.
5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Python.
 - Pour Image, sélectionnez `aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`.
 - Pour Rôle de service, laissez l'option Nouveau rôle de service sélectionnée. Notez le nom du rôle. Cela sera nécessaire lorsque vous mettrez à jour les autorisations IAM du projet ultérieurement dans cet exemple.
6. Choisissez Créer un projet de génération.
7. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
8. Dans le volet de navigation, choisissez Rôles et sélectionnez le rôle de service associé à votre projet. Vous pouvez trouver votre rôle dans le projet en CodeBuild sélectionnant votre projet de construction, en choisissant Modifier, Environnement, puis Rôle de service.
9. Sélectionnez l'onglet Trust relationships (Relations d'approbation), puis Edit trust policy (Modifier la politique d'approbation).
10. Ajoutez la politique intégrée suivante à votre rôle IAM. Cela sera utilisé pour déployer votre AWS SAM infrastructure ultérieurement. Pour plus d'informations, consultez [Ajout et suppression d'autorisations basées sur l'identité IAM](#) dans le Guide de l'utilisateur IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateLambdaPermissions",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
```

```
        """
    ]
}
}
```

Configurer le buildspec du projet

Afin de mettre à jour la fonction Lambda, le script lit les variables d'environnement depuis le buildspec pour trouver le nom, le nom de la variable d'environnement et la valeur de la variable d'environnement de la fonction Lambda.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Passer à l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
env:
  variables:
    LAMBDA_FUNC_NAME: "<lambda-function-name>"
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
    LAMBDA_ENV_VARIABLE_VALUE: "true"
  phases:
    install:
      commands:
        - pip3 install boto3
    build:
      commands:
        - python3 update_lambda_environment_variables.py
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Mettez à jour votre configuration Lambda

Utilisez CodeBuild Lambda Python pour mettre à jour automatiquement la configuration de votre fonction Lambda.

Pour mettre à jour la configuration de votre fonction Lambda

1. Choisissez Démarrer la génération.
2. Une fois le build terminé, accédez à votre fonction Lambda.
3. Sélectionnez Configuration, puis Variables d'environnement. Vous devriez voir une nouvelle variable d'environnement contenant une clé `FEATURE_ENABLED` et une valeur `true`.

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées pour votre CodeBuild projet.

Pour nettoyer votre infrastructure

1. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.
2. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.
3. Si vous avez créé une fonction Lambda aux fins de cet exemple, choisissez Actions et fonction Supprimer pour nettoyer votre fonction Lambda.

Extensions

Si vous souhaitez étendre cet exemple pour gérer d'autres AWS ressources à l'aide de AWS CodeBuild Lambda Python :

- Mettez à jour le script Python pour modifier les nouvelles ressources à l'aide de Boto3.
- Mettez à jour le rôle IAM associé à votre CodeBuild projet afin d'obtenir des autorisations pour les nouvelles ressources.
- Ajoutez toutes les nouvelles variables d'environnement associées aux nouvelles ressources à vos spécifications de construction.

Exécutez des builds sur des flottes à capacité réservée

CodeBuild propose les flottes de calcul suivantes :

- Flottes à la demande
- Flottes de capacité réservée

Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites à la fin de la construction. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.

 Note

Les flottes à la demande ne sont pas compatibles avec macOS.

CodeBuild propose également des flottes de capacités réservées contenant des instances alimentées par Amazon EC2 et gérées par CodeBuild. Avec les flottes de capacité réservée, vous configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

 Important

Quelle que soit la durée pendant laquelle vous exécutez une instance, les flottes de capacité réservée sont facturées initialement par instance, après quoi des coûts supplémentaires peuvent être associés. Pour de plus amples informations, veuillez consulter <https://aws.amazon.com/codebuild/pricing/>.

Rubriques

- [Création d'un parc de capacités réservées](#)
- [Bonnes pratiques](#)
- [Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ?](#)
- [Comment fonctionne le calcul basé sur les attributs ?](#)
- [Puis-je spécifier manuellement une instance Amazon EC2 pour mon parc ?](#)
- [Quelles régions prennent en charge les flottes à capacité réservée ?](#)
- [Comment configurer une flotte de macOS à capacité réservée ?](#)

- [Comment configurer une Amazon Machine Image \(AMI\) personnalisée pour un parc de capacités réservées ?](#)
- [Limitations des flottes à capacité réservée](#)
- [Propriétés du parc à capacité réservée](#)
- [Échantillons de capacité réservée avec AWS CodeBuild](#)

Création d'un parc de capacités réservées

Suivez les instructions ci-dessous pour créer un parc de capacités réservées.

Pour créer un parc de capacités réservées

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Compute fleets, puis Create Fleet.
3. Dans le champ de texte Calculer le nom du parc, saisissez le nom de votre parc.
4. Dans le menu déroulant Système d'exploitation, sélectionnez le système d'exploitation.
5. Dans le menu déroulant Architecture, sélectionnez l'architecture.
6. (Facultatif) Sélectionnez Utiliser le mode d'exécution de l'instance. Facultatif pour exécuter directement sur une instance Amazon EC2 au lieu d'un conteneur Docker. Choisissez ensuite une version majeure et une version mineure.
7. (Facultatif) Dans Configuration supplémentaire, procédez comme suit :
 - Sélectionnez Configurer le VPC - facultatif pour connecter votre flotte à un VPC afin d'accéder à des ressources privées pendant l'utilisation.
 - Dans le menu déroulant VPC, sélectionnez un VPC auquel votre CodeBuild flotte aura accès.
 - Dans le menu déroulant Sous-réseaux, sélectionnez les sous-réseaux à utiliser CodeBuild pour configurer votre configuration VPC.
 - Dans le menu déroulant Groupes de sécurité, sélectionnez les groupes de sécurité CodeBuild à utiliser pour fonctionner avec votre VPC.
 - Dans le champ Rôle de service de flotte, choisissez un rôle de service existant.

 Note

Assurez-vous que votre rôle dans la flotte dispose des autorisations nécessaires. Pour de plus amples informations, veuillez consulter [Autoriser un utilisateur à ajouter une politique d'autorisation pour un rôle de service de flotte](#).

- Si vous avez choisi le système d'exploitation Amazon Linux, sélectionnez Définir les configurations de proxy (facultatif) pour appliquer le contrôle d'accès réseau à vos instances à capacité réservée.
- Pour Comportement par défaut, choisissez d'autoriser ou de refuser le trafic sortant vers toutes les destinations par défaut.
- Pour les règles de proxy, choisissez Ajouter une règle de proxy pour spécifier les domaines de destination ou IPs pour autoriser ou refuser le contrôle d'accès au réseau.
- Sélectionnez Configurer une AMI personnalisée. Cette option est facultative pour utiliser une Amazon Machine Image (AMI) personnalisée.
- Dans le menu déroulant AMI, sélectionnez une Amazon Machine Image (AMI) pour votre flotte.
- Dans le champ Rôle de service de flotte, choisissez un rôle de service existant.

 Note

Assurez-vous que votre rôle dans la flotte dispose des autorisations nécessaires. Pour de plus amples informations, veuillez consulter [Autoriser un utilisateur à ajouter une politique d'autorisation pour un rôle de service de flotte](#).

8. Dans Configuration de la capacité, dans Mode de sélection de calcul, choisissez l'une des options suivantes :

- Si vous choisissez Sélection guidée, procédez comme suit :
 - Pour Compute, choisissez le type d'instances inclus dans ce parc.
 - Dans le champ de texte Capacité, entrez le nombre minimum d'instances dans le parc.
 - (Facultatif) Dans Configuration supplémentaire, procédez comme suit :
 - Sélectionnez Configurer le dimensionnement : facultatif pour redimensionner automatiquement votre flotte en fonction de cette configuration. Dans le menu déroulant

optionnel Mode de mise à l'échelle, choisissez le comportement à adopter lorsque la demande dépasse la capacité de la flotte.

- Si vous choisissez Instance personnalisée, procédez comme suit :
 - Dans le menu déroulant Calculer le type d'instance, sélectionnez le type d'instances inclus dans ce parc.
 - Dans le champ de texte Taille de volume EBS supplémentaire - facultatif, entrez le volume supplémentaire aux 64 Go d'espace disque fournis.
 - Dans le champ de texte Capacité, entrez le nombre minimum d'instances dans le parc.
 - (Facultatif) Dans Configuration supplémentaire, procédez comme suit :
 - Sélectionnez Configurer le dimensionnement : facultatif pour redimensionner automatiquement votre flotte en fonction de cette configuration. Dans le menu déroulant optionnel Mode de mise à l'échelle, choisissez le comportement à adopter lorsque la demande dépasse la capacité de la flotte.
9. Choisissez Créer un parc informatique.
10. Une fois le parc informatique créé, créez un nouveau CodeBuild projet ou modifiez un projet existant. Dans Environnement, choisissez Capacité réservée sous Modèle de provisionnement, puis choisissez le parc spécifié sous Nom du parc.

Bonnes pratiques

Lorsque vous utilisez des flottes de capacité réservée, nous vous recommandons de suivre ces bonnes pratiques.

- Nous vous recommandons d'utiliser le mode de cache source pour améliorer les performances de génération en mettant en cache la source.
- Nous vous recommandons d'utiliser la mise en cache des couches Docker pour améliorer les performances de création en mettant en cache les couches Docker existantes.

Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ?

Oui, vous pouvez optimiser l'utilisation de la capacité d'une flotte en l'utilisant dans le cadre de plusieurs projets.

Important

Lorsque vous utilisez la fonctionnalité de capacité réservée, les données mises en cache sur les instances de flotte, y compris les fichiers source, les couches Docker et les répertoires mis en cache spécifiés dans les spécifications de construction, peuvent être accessibles aux autres projets du même compte. Cela est intentionnel et permet aux projets au sein d'un même compte de partager des instances de flotte.

Comment fonctionne le calcul basé sur les attributs ?

Si vous choisissez `ATTRIBUTE_BASED_COMPUTE` celui de votre `flottecomputeType`, vous pouvez spécifier les attributs dans un nouveau champ appelé `computeConfiguration`. Ces attributs incluent vCPUs, la mémoire, l'espace disque et le `machineType`. C'`machineType` est l'un `GENERAL` ou l'autre `NVME`. Après avoir spécifié un ou plusieurs des attributs disponibles, il CodeBuild choisira un type de calcul parmi les types d'instances pris en charge disponibles une fois finalisé `computeConfiguration`.

Note

CodeBuild choisira l'instance la moins chère qui répond à toutes les exigences de saisie. La mémoire, le v CPUs et l'espace disque des instances choisies seront tous supérieurs ou égaux aux exigences d'entrée. Vous pouvez vérifier les problèmes résolus `computeConfiguration` dans le parc créé ou mis à jour.

Si vous saisissez un `computeConfiguration` qu'il n'est pas possible de satisfaire CodeBuild, vous recevrez une exception de validation. Notez également que le comportement de dépassement de flotte à la demande sera remplacé par le comportement de file d'attente s'il n'`computeConfiguration` est pas disponible pour la demande.

Puis-je spécifier manuellement une instance Amazon EC2 pour mon parc ?

Oui, vous pouvez saisir directement l'instance Amazon EC2 souhaitée dans la console en sélectionnant Instance personnalisée ou en configurant le paramètre API, `InstanceType`. Ce champ est utilisé dans les cas suivants APIs : `CreateFleet`, `UpdateFleet`, `CreateProject`, `UpdateProject` et `StartBuild`. Pour de plus amples informations, veuillez consulter [Compute instance type](#).

Quelles régions prennent en charge les flottes à capacité réservée ?

Capacité réservée Les flottes Amazon Linux et Windows sont prises en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort), Europe (Irlande) et Amérique du Sud (São Paulo). Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Capacité réservée Les flottes macOS Medium sont prises en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Sydney) et Europe (Francfort). **Capacité réservée** Les grandes flottes macOS sont prises en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon) et Asie-Pacifique (Sydney).

Comment configurer une flotte de macOS à capacité réservée ?

Pour configurer un parc macOS à capacité réservée

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Compute fleets, puis Create Fleet.
3. Dans le champ de texte Calculer le nom du parc, saisissez le nom de votre parc.
4. Dans le menu déroulant Système d'exploitation, sélectionnez macOS.
5. Dans le champ Calcul, sélectionnez l'un des types de machines de calcul suivants : Apple M2, 24 Go de mémoire, 8 V CPUs ou Apple M2, 32 Go de mémoire, 12 CPUs V.
6. Dans le champ de texte Capacité, entrez le nombre minimum d'instances dans le parc.
7. (Facultatif) Pour utiliser une image personnalisée pour votre flotte [Comment configurer une Amazon Machine Image \(AMI\) personnalisée pour un parc de capacités réservées ?](#), assurez-vous que votre Amazon Machine Image (AMI) répond aux exigences requises.
8. (Facultatif) Pour configurer un VPC avec votre flotte, dans Configuration supplémentaire, procédez comme suit :
 - Dans le menu déroulant VPC - facultatif, sélectionnez un VPC auquel votre CodeBuild flotte aura accès.
 - Dans le menu déroulant Sous-réseaux, sélectionnez les sous-réseaux à utiliser CodeBuild pour configurer votre configuration VPC.

- Dans le menu déroulant Groupes de sécurité, sélectionnez les groupes de sécurité CodeBuild à utiliser pour fonctionner avec votre VPC.
- Dans le champ Rôle de service de flotte, choisissez un rôle de service existant.

 Note

Assurez-vous que votre rôle dans la flotte dispose des autorisations nécessaires. Pour de plus amples informations, veuillez consulter [Autoriser un utilisateur à ajouter une politique d'autorisation pour un rôle de service de flotte](#).

9. Choisissez Créer une flotte de calcul et attendez que l'instance de flotte soit lancée. Une fois lancée, la capacité sera n/n , où n est la capacité fournie ?
10. Une fois le parc informatique lancé, créez un nouveau CodeBuild projet ou modifiez un projet existant. Dans Environnement, choisissez Capacité réservée sous Modèle de provisionnement, puis choisissez le parc spécifié sous Nom du parc.

Comment configurer une Amazon Machine Image (AMI) personnalisée pour un parc de capacités réservées ?

Pour configurer une Amazon Machine Image (AMI) personnalisée pour un parc de capacités réservées

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Compute fleets, puis Create Fleet.
3. Dans le champ de texte Calculer le nom du parc, saisissez le nom de votre parc.
4. Choisissez une image personnalisée pour votre flotte et assurez-vous que votre Amazon Machine Image (AMI) répond aux exigences suivantes :
 - Si votre type d'environnement est MAC_ARM le cas, assurez-vous que votre architecture AMI est 64 bitsMac - Arm.
 - Si votre type d'environnement est LINUX_EC2 le cas, assurez-vous que votre architecture AMI est 64 bitsx86.
 - Si votre type d'environnement est ARM_EC2 le cas, assurez-vous que votre architecture AMI est 64 bitsArm.

- Si votre type d'environnement est `WINDOWS_EC2` le cas, assurez-vous que votre architecture AMI est 64 bitsx86.
- L'AMI autorise le CodeBuild service Organization ARN. Pour une liste des organisations ARNs, voir [Amazon Machine Images \(AMI\)](#).
- Si l'AMI est chiffrée à l'aide d'une AWS KMS clé, AWS KMS celle-ci doit également autoriser l'ID d'organisation du CodeBuild service. Pour une liste des organisations IDs, voir [Amazon Machine Images \(AMI\)](#). Pour plus d'informations sur AWS KMS les clés, consultez [Autoriser les organisations et OUs utiliser une clé KMS](#) dans le guide de l'utilisateur Amazon EC2. Pour autoriser CodeBuild l'organisation à utiliser une clé KMS, ajoutez la déclaration suivante à la politique de clé :

```
{
  "Sid": "Allow access for organization root",
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
    "kms:Describe*",
    "kms:List*",
    "kms:Get*",
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalOrgID": "o-123example"
    }
  }
}
```

- Dans le champ Rôle du service de flotte, accordez les autorisations Amazon EC2 suivantes :
JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeImages",
            "ec2:DescribeSnapshots"
        ],
        "Resource": "*"
    }
}
```

Limitations des flottes à capacité réservée

Il existe certains cas d'utilisation que les flottes de capacité réservée ne prennent pas en charge, et s'ils vous concernent, utilisez plutôt des flottes à la demande :

- Les flottes de capacité réservée ne prennent pas en charge les indicateurs d'utilisation des builds.
- Les flottes de macOS à capacité réservée ne prennent pas en charge les sessions de débogage.

Pour plus d'informations sur les limites et les quotas, consultez [Flottes informatiques](#).

Propriétés du parc à capacité réservée

Un parc à capacité réservée contient les propriétés suivantes. Pour plus d'informations sur les flottes de capacité réservée, consultez [Exécutez des builds sur des flottes à capacité réservée](#).

Système d'exploitation

Le système d'exploitation. Les systèmes d'exploitation suivants sont disponibles :

- Amazon Linux
- macOS
- Windows Server 2019
- Windows Server 2022

Architecture

L'architecture du processeur. Les architectures disponibles sont les suivantes :

- x86_64
- Bras 64

Type d'environnement

Les types d'environnement disponibles lorsque Amazon Linux est sélectionné. Les types d'environnement suivants sont disponibles :

- Linux EC2
- GPU Linux

Type d'instance de calcul

Les configurations de calcul pour les instances de flotte.

Sélection guidée

Spécifiez différents types de calcul en sélectionnant les paramètres du vCPU, de la mémoire et de l'espace disque. Pour plus d'informations sur la disponibilité des types de calcul par région, consultez [À propos des types d'environnement de capacité réservée](#).

Instance personnalisée

Spécifiez manuellement le type d'instance souhaité.

Capacity

Le nombre initial de machines allouées au parc, qui définit le nombre de builds pouvant être exécutés en parallèle.

Comportement de débordement

Définit le comportement lorsque le nombre de builds dépasse la capacité de la flotte.

À la demande

Les versions Overflow s'exécutent CodeBuild à la demande.

Note

Si vous choisissez de définir votre comportement de débordement à la demande lors de la création d'une flotte connectée au VPC, assurez-vous d'ajouter les autorisations VPC requises à votre rôle de service de projet. Pour plus d'informations, consultez [Exemple de déclaration de politique autorisant CodeBuild l'accès aux AWS services nécessaires à la création d'une interface réseau VPC](#).

⚠ Important

Si vous choisissez de définir votre comportement de débordement à la demande, notez que les versions de débordement seront facturées séparément, comme pour Amazon EC2 à la demande. Pour de plus amples informations, veuillez consulter <https://aws.amazon.com/codebuild/pricing/>.

File d'attente

Les cycles de build sont placés dans une file d'attente jusqu'à ce qu'une machine soit disponible. Cela limite les coûts supplémentaires car aucune machine supplémentaire n'est allouée.

Amazon Machine Images (AMI)

Les propriétés Amazon Machine Image (AMI) de votre flotte. Les propriétés suivantes sont prises en charge par CodeBuild :

Régions AWS	ARN de l'organisation	Organization ID
us-east-1	arn:aws:organizations::851725618577:organization/o-c6wcu152r1	o-c6wcu152r1
us-east-2	arn:aws:organizations::992382780434:organization/o-seufr2suvq	o-seufr2suvq
us-west-2	arn:aws:organizations::381491982620:organization/o-0412o99a4r	o-0412o99a4r
ap-northeast-1	arn:aws:organizations::891376993293:	o-b6k3sjqavm

Régions AWS	ARN de l'organisation	Organization ID
	organization/o-b6k3sjqavm	
ap-south-1	arn:aws:organizations::891376924779:organization/o-krtah1lkeg	o-krtah1lkeg
ap-southeast-1	arn:aws:organizations::654654522137:organization/o-mcn8uvc3tp	o-mcn8uvc3tp
ap-southeast-2	arn:aws:organizations::767398067170:organization/o-6crt0f6bu4	o-6crt0f6bu4
eu-central-1	arn:aws:organizations::590183817084:organization/o-lb2lne3te6	o-lb2lne3te6
eu-west-1	arn:aws:organizations::891376938588:organization/o-ullrrg5qf0	o-ullrrg5qf0
sa-east-1	arn:aws:organizations::533267309133:organization/o-db63c45ozw	o-db63c45ozw

Configuration supplémentaire

VPC - en option

Le VPC auquel votre CodeBuild flotte aura accès. Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Note

Si une dérogation de flotte est spécifiée lors de l'appel de l' `StartBuild` API, la configuration VPC du projet CodeBuild sera ignorée.

Sous-réseaux

Les sous-réseaux VPC CodeBuild utilisés pour configurer votre configuration VPC. Notez que les flottes de capacité réservée ne prennent en charge qu'un seul sous-réseau dans une seule zone de disponibilité. Assurez-vous également que vos sous-réseaux incluent une passerelle NAT.

Groupes de sécurité

Les groupes de sécurité VPC CodeBuild utilisés avec votre VPC. Assurez-vous que vos groupes de sécurité autorisent les connexions sortantes.

Rôle du service de flotte

Définit le rôle de service de votre flotte à partir d'un rôle de service existant dans votre compte.

Définir les configurations de proxy (facultatif)

Configurations de proxy qui appliquent un contrôle d'accès réseau à vos instances de capacité réservée. Pour de plus amples informations, veuillez consulter [Utilisation AWS CodeBuild avec un serveur proxy géré](#).

Note

Les configurations de proxy ne sont pas compatibles avec VPC, Windows ou macOS.

Comportement par défaut

Définit le comportement du trafic sortant.

Allow

Autorise le trafic sortant vers toutes les destinations par défaut.

Refuser

Refuse le trafic sortant vers toutes les destinations par défaut.

Règles relatives au proxy

Spécifie les domaines de destination ou IPs pour autoriser ou refuser le contrôle d'accès au réseau.

Échantillons de capacité réservée avec AWS CodeBuild

Ces échantillons peuvent être utilisés pour expérimenter avec des flottes de capacité réservée dans CodeBuild.

Rubriques

- [Exemple de mise en cache avec capacité réservée](#)

Exemple de mise en cache avec capacité réservée

Un cache peut stocker les éléments réutilisables de votre environnement de génération et les utiliser sur plusieurs générations. Cet exemple montre comment activer la mise en cache dans votre projet de construction en utilisant une capacité réservée. Pour de plus amples informations, veuillez consulter [Des mises en cache pour améliorer les performances](#).

Vous pouvez commencer par spécifier un ou plusieurs modes de cache dans les paramètres de votre projet :

Cache :

Type: LOCAL

Modes :

- LOCAL_CUSTOM_CACHE
- LOCAL_DOCKER_LAYER_CACHE
- LOCAL_SOURCE_CACHE

 Note

Assurez-vous d'activer le mode privilégié afin d'utiliser le cache de couche Docker.

Les paramètres buildspec de votre projet doivent ressembler à ce qui suit :

```
version: 0.2
  phases:
    build:
      commands:
        - echo testing local source cache
        - touch /codebuild/cache/workspace/foobar.txt
        - git checkout -b cached_branch
        - echo testing local docker layer cache
        - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
        - echo testing local custom cache
        - touch foo
        - mkdir bar && ln -s foo bar/foo2
        - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
        - "[ -f foo ] || exit 1"
        - "[ -L bar/foo2 ] || exit 1"
        - "[ -f bar/bar/foo3 ] || exit 1"
        - "[ -f bar/bar/foo4 ] || exit 1"
      cache:
        paths:
          - './foo'
          - './bar/**/*'
          - './bar/bar/foo3'
```

Vous pouvez commencer par exécuter une compilation avec le nouveau projet pour démarrer le cache. Une fois cette opération terminée, vous devez commencer une autre version avec une spécification de construction alternative, similaire à ce qui suit :

```
version: 0.2
  phases:
    build:
      commands:
        - echo testing local source cache
        - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
        - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
```

```
- echo testing local docker layer cache
- docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
- echo testing local custom cache
- "[ -f foo ] || exit 1"
- "[ -L bar/foo2 ] || exit 1"
- "[ -f bar/bar/foo3 ] || exit 1"
- "[ -f bar/bar/foo4 ] || exit 1"
cache:
  paths:
    - './foo'
    - './bar/**/*'
    - './bar/bar/foo3'
```

Exécuter des builds par lots

Vous pouvez l'utiliser AWS CodeBuild pour exécuter des versions simultanées et coordonnées d'un projet avec des versions par lots.

Rubriques

- [Rôle de sécurité](#)
- [Types de build par lots](#)
- [Mode de rapport par lots](#)
- [En savoir plus](#)

Rôle de sécurité

Les générations en lot introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est obligatoire car CodeBuild il doit être capable d'appeler les RetryBuild actions StartBuildStopBuild, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même rôle qu'ils utilisent dans leur build, pour deux raisons :

- Donner au rôle de build StartBuild, StopBuild et RetryBuild les autorisations permettrait à une seule build de lancer davantage de builds via le buildspec.
- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds elles-mêmes puissent contourner ces restrictions.

Types de build par lots

CodeBuild prend en charge les types de génération par lots suivants :

Types de build par lots

- [Construire un graphe](#)
- [Construire une liste](#)
- [Construire une matrice](#)
- [Construire un ventilateur](#)

Construire un graphe

Un graphe de génération définit un ensemble de tâches qui dépendent d'autres tâches du lot.

L'exemple suivant définit un graphe de construction qui crée une chaîne de dépendances.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
```

```
env:
  fleet: fleet_name
```

Dans cet exemple :

- build1s'exécute en premier car il n'a aucune dépendance.
- build2dépend debuild1, donc build2 s'exécute une fois build1 terminé.
- build3dépend debuild2, donc build3 s'exécute une fois build2 terminé.

Pour plus d'informations sur la syntaxe buildspec du graphe de construction, consultez. [batch/build-graph](#)

Construire une liste

Une liste de build définit un certain nombre de tâches exécutées en parallèle.

L'exemple suivant définit une liste de build. Les build2 builds build1 et seront exécutés en parallèle.

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
```

```
compute-type: GENERAL_LINUX_XLAGRE
```

Pour plus d'informations sur la syntaxe buildspec de la liste de compilation, consultez. [batch/build-list](#)

Construire une matrice

Une matrice de génération définit les tâches avec différentes configurations qui s'exécutent en parallèle. CodeBuild crée une version distincte pour chaque combinaison de configuration possible.

L'exemple suivant montre une matrice de construction avec deux fichiers buildspec et trois valeurs pour une variable d'environnement.

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Dans cet exemple, CodeBuild crée six versions :

- matrix1.yml avec \$MY_VAR=VALUE1
- matrix1.yml avec \$MY_VAR=VALUE2
- matrix1.yml avec \$MY_VAR=VALUE3
- matrix2.yml avec \$MY_VAR=VALUE1
- matrix2.yml avec \$MY_VAR=VALUE2
- matrix2.yml avec \$MY_VAR=VALUE3

Chaque build comportera les paramètres suivants :

- `ignore-failur` réglé sur `false`
- `env/typer` réglé sur `LINUX_CONTAINER`
- `env/imager` réglé sur `aws/codebuild/amazonlinux-x86_64-standard:4.0`
- `env/privileged-mod` réglé sur `true`

Ces builds s'exécutent en parallèle.

Pour plus d'informations sur la syntaxe `buildspec` de la matrice de construction, consultez [batch/build-matrix](#)

Construire un ventilateur

Un `fanout` de build définit une tâche qui sera divisée en plusieurs builds dans le lot. Cela peut être utilisé pour exécuter des tests en parallèle. CodeBuild crée une version distincte pour chaque fragment de cas de test en fonction de la valeur définie dans le `parallelism` champ.

L'exemple suivant définit un `fanout` de build qui crée cinq builds exécutés en parallèle.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'equal-distribution'
```

Dans cet exemple, en supposant que 100 tests doivent être exécutés, CodeBuild crée cinq versions qui exécutent chacune 20 tests en parallèle.

Pour plus d'informations sur la syntaxe buildspec du graphe de construction, consultez [batch/build-fanout](#)

Mode de rapport par lots

Si le fournisseur source de votre projet est Bitbucket ou GitHub Enterprise GitHub, et que votre projet est configuré pour signaler les statuts de construction au fournisseur source, vous pouvez sélectionner la manière dont vous souhaitez que les statuts de génération par lots soient envoyés au fournisseur source. Vous pouvez choisir d'envoyer les statuts sous forme de rapport d'état agrégé unique pour le lot, ou de faire en sorte que le statut de chaque build du lot soit signalé individuellement.

Pour plus d'informations, consultez les rubriques suivantes :

- [Configuration par lots \(création\)](#)
- [Configuration par lots \(mise à jour\)](#)

En savoir plus

Pour plus d'informations, consultez les rubriques suivantes :

- [Référence Batch Build Buildspec](#)
- [Configuration par lots](#)
- [Exécuter une compilation par lots \(AWS CLI\)](#)
- [Arrêter les compilations par lots AWS CodeBuild](#)

Exécuter des tests parallèles dans des builds par lots

Vous pouvez l'utiliser AWS CodeBuild pour exécuter des tests parallèles dans des builds par lots. L'exécution parallèle de tests est une approche de test dans laquelle plusieurs cas de test s'exécutent simultanément sur différents environnements, machines ou navigateurs, plutôt que de s'exécuter de manière séquentielle. Cette approche permet de réduire considérablement le temps global d'exécution des tests et d'améliorer l'efficacité des tests. Dans CodeBuild, vous pouvez répartir vos tests sur plusieurs environnements et les exécuter simultanément.

Les principaux avantages de l'exécution de tests en parallèle sont les suivants :

1. Temps d'exécution réduit : les tests qui dureraient plusieurs heures de manière séquentielle peuvent être achevés en quelques minutes.
2. Meilleure utilisation des ressources : utilise efficacement les ressources informatiques disponibles.
3. Feedback anticipé - L'achèvement plus rapide des tests signifie un retour d'information plus rapide pour les développeurs.
4. Rentable : permet d'économiser du temps et des coûts informatiques à long terme.

Lors de la mise en œuvre de l'exécution de tests en parallèle, deux approches principales sont généralement envisagées : les environnements séparés et le multithreading. Bien que les deux méthodes visent à réaliser des tests simultanément, elles diffèrent considérablement en termes de mise en œuvre et d'efficacité. Les environnements distincts créent des instances isolées dans lesquelles chaque suite de tests s'exécute indépendamment, tandis que le multithreading exécute plusieurs tests simultanément dans le même espace de processus en utilisant différents threads.

Les principaux avantages des environnements séparés par rapport au multithreading sont les suivants :

1. Isolation - Chaque test s'exécute dans un environnement complètement isolé, évitant ainsi toute interférence entre les tests.
2. Conflits de ressources : aucune concurrence pour les ressources partagées, ce qui se produit souvent dans le multithreading.
3. Stabilité : moins sujet aux conditions de course et aux problèmes de synchronisation.
4. Débogage facilité : lorsque les tests échouent, il est plus simple d'en identifier la cause car chaque environnement est indépendant.
5. Gestion des états : gérez facilement les problèmes d'état partagés qui entravent les tests multithread.
6. Meilleure évolutivité - Possibilité d'ajouter facilement d'autres environnements sans complexité.

Rubriques

- [Support dans AWS CodeBuild](#)
- [Activer l'exécution de tests en parallèle dans les versions par lots](#)
- [Utilisez la codebuild-tests-run commande CLI](#)

- [Utilisez la codebuild-glob-search commande CLI](#)
- [À propos du fractionnement des tests](#)
- [Fusionnez automatiquement les rapports de construction individuels](#)
- [Exemple d'exécution de tests en parallèle pour différents frameworks de test](#)

Support dans AWS CodeBuild

AWS CodeBuild fournit un support robuste pour l'exécution de tests en parallèle grâce à sa fonctionnalité de génération par lots, spécialement conçue pour tirer parti de l'exécution d'environnements séparés. Cette mise en œuvre s'aligne parfaitement sur les avantages des environnements de test isolés.

Construction par lots avec distribution de tests

CodeBuild la fonctionnalité de génération par lots permet de créer plusieurs environnements de génération qui s'exécutent simultanément. Chaque environnement fonctionne comme une unité complètement isolée, avec ses propres ressources informatiques, son propre environnement d'exécution et ses propres dépendances. Grâce à la configuration de génération par lots, vous pouvez spécifier le nombre d'environnements parallèles dont ils ont besoin et la manière dont les tests doivent être répartis entre eux.

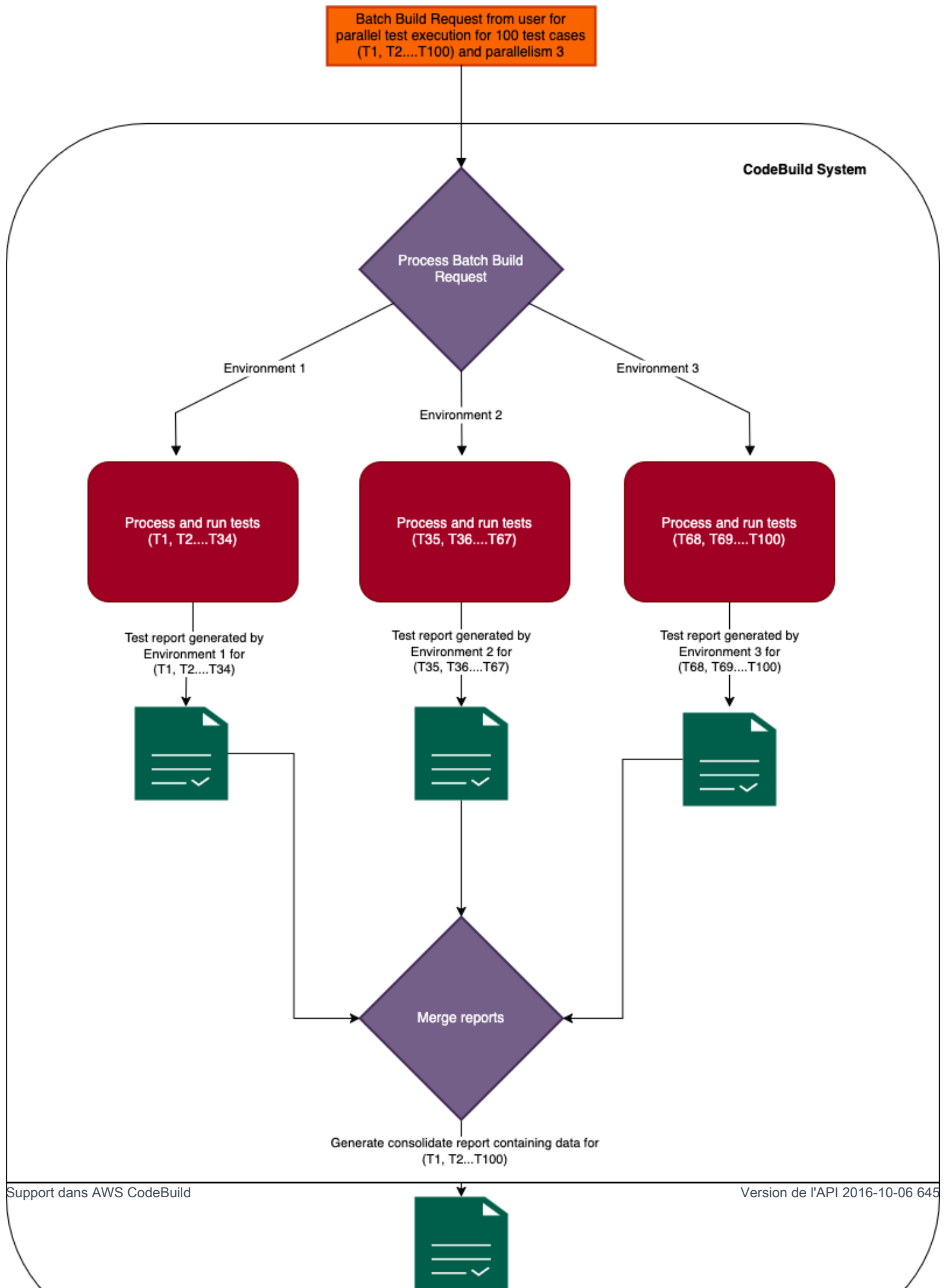
Tester la CLI de partitionnement

CodeBuild inclut un mécanisme de distribution de tests intégré via son outil CLI `codebuild-tests-run`, qui divise automatiquement les tests en différents environnements.

Agrégation de rapports

L'un des principaux points forts de la mise en œuvre CodeBuild est sa capacité à gérer l'agrégation des résultats de test de manière fluide. Lorsque les tests s'exécutent dans des environnements distincts, collecte et combine CodeBuild automatiquement les rapports de test de chaque environnement dans un rapport de test unifié au niveau de la génération par lots. Cette consolidation fournit une vue complète des résultats des tests tout en préservant les avantages d'efficacité de l'exécution parallèle.

Le schéma suivant explique le concept complet de l'exécution de tests en parallèle dans AWS CodeBuild.



Activer l'exécution de tests en parallèle dans les versions par lots

Pour exécuter des tests en parallèle, mettez à jour le fichier buildspec de compilation par lots pour inclure le champ `build-fanout` et le nombre de builds parallèles pour diviser la suite de tests dans le champ, comme indiqué ci-dessous. `parallelism` Le `parallelism` champ indique le nombre d'exécuteurs indépendants configurés pour exécuter la suite de tests.

Pour exécuter les tests dans plusieurs environnements d'exécution parallèle, définissez le `parallelism` champ sur une valeur supérieure à zéro. Dans l'exemple ci-dessous, `parallelism` est défini sur cinq, ce qui signifie qu'il CodeBuild démarre cinq versions identiques qui exécutent une partie de la suite de tests en parallèle.

Vous pouvez utiliser la commande [codebuild-tests-run](#) CLI pour diviser et exécuter vos tests. Vos fichiers de test seront divisés et une partie de vos tests sera exécutée dans chaque version. Cela réduit le temps total nécessaire à l'exécution de la suite de tests complète. Dans l'exemple suivant, les tests seront divisés en cinq et les points de division seront calculés en fonction du nom des tests.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install jest-junit --save-dev
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/_tests_/**/*test.js'" \
          --sharding-strategy 'equal-distribution'

  post_build:
    commands:
```

```
- codebuild-glob-search '**/*.xml'
- echo "Running post-build steps..."
- echo "Build completed on `date`"

reports:
  test-reports:
    files:
      - '**/junit.xml'
    base-directory: .
    discard-paths: yes
    file-format: JUNITXML
```

Si les rapports sont configurés pour la version build-fanout, les rapports de test sont générés séparément pour chaque version, qui peuvent être consultés sous l'onglet Rapports des versions correspondantes dans la console. AWS CodeBuild

Pour plus d'informations sur la façon d'exécuter des tests parallèles par lots, consultez [Exemple d'exécution de tests en parallèle pour différents frameworks de test](#).

Utilisez la **codebuild-tests-run** commande CLI

AWS CodeBuild fournit une CLI qui prendra la commande de test et l'emplacement du fichier de test en entrée. La CLI avec ces entrées divisera les tests en un nombre de partitions tel que spécifié dans le `parallelism` champ en fonction des noms de fichiers de test. L'attribution des fichiers de test au shard est décidée par la stratégie de sharding.

```
codebuild-tests-run \
  --files-search "codebuild-glob-search '**/__tests__/*.js'" \
  --test-command 'npx jest --runInBand --coverage' \
  --sharding-strategy 'equal-distribution'
```

Le tableau suivant décrit les champs de la commande `codebuild-tests-run` CLI.

Nom de champ	Type	Obligatoire ou facultatif	Définition
<code>test-command</code>	String	Obligatoire	Cette commande est utilisée pour exécuter les tests.

Nom de champ	Type	Obligatoire ou facultatif	Définition
files-search	String	Obligatoire	Cette commande fournit une liste de fichiers de test. Vous pouvez utiliser la commande codebuild -glob-search CLI AWS CodeBuild fournie ou tout autre outil de recherche de fichiers de votre choix.  Note Assurez-vous que la files-search commande affiche des noms de fichiers, séparés chacun par une nouvelle ligne.

Nom de champ	Type	Obligatoire ou facultatif	Définition
sharding-strategy	Enum	Facultatif	Valeurs valides : equal-distribution (par défaut), stability • equal-distribution : répartissez les fichiers de test de manière uniforme en fonction des noms de fichiers de test. • stability : partagez des fichiers de test en utilisant un hachage cohérent des noms de fichiers. Pour de plus amples informations, veuillez consulter À propos du fractionnement des tests.

La `codebuild-tests-run` CLI fonctionne d'abord pour identifier la liste des fichiers de test à l'aide de la commande fournie dans le `files-search` paramètre. Il détermine ensuite un sous-ensemble de fichiers de test désignés pour la partition actuelle (environnement) en utilisant la stratégie de partitionnement spécifiée. Enfin, ce sous-ensemble de fichiers de test est formaté sous forme de liste séparée par des espaces et ajouté à la fin de la commande fournie dans le `test-command` paramètre avant d'être exécuté.

Pour les frameworks de test qui n'acceptent pas les listes séparées par des espaces, la `codebuild-tests-run` CLI fournit une alternative flexible via la variable d'environnement `CODEBUILD_CURRENT_SHARD_FILES`. Cette variable contient une liste séparée par de nouvelles lignes de chemins de fichiers de test désignés pour la partition de build actuelle. En tirant parti de cette variable d'environnement, vous pouvez facilement vous adapter aux différentes exigences du framework de test, en prenant en compte ceux qui attendent des formats d'entrée différents de ceux des listes séparées par des espaces. De plus, vous pouvez également formater les noms des fichiers de test selon les besoins du framework de test. Voici un exemple d'utilisation de `CODEBUILD_CURRENT_SHARD_FILES` sous Linux avec le framework Django. Voici ce qui est utilisé pour obtenir les chemins de fichiers de notation par points supportés par Django :

```
codebuild-tests-run \
  -files-search "codebuild-glob-search '/tests/test.py'" \
  -test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
-E "s/\//_/g; s/\.py$/./; s/_/./g")' \
  -sharding-strategy 'equal-distribution'
```

Note

Notez que la variable d'environnement `CODEBUILD_CURRENT_SHARD_FILES` ne peut être utilisée que dans le cadre de la `codebuild-tests-run` CLI. De plus, si vous utilisez la commande de test interne `CODEBUILD_CURRENT_SHARD_FILES`, mettez-la entre guillemets, comme indiqué dans l'exemple ci-dessus.

Utilisez la **codebuild-glob-search** commande CLI

AWS CodeBuild fournit un outil CLI intégré appelé `codebuild-glob-search` qui vous permet de rechercher des fichiers dans votre répertoire de travail en fonction d'un ou de plusieurs modèles globaux. Cet outil peut être particulièrement utile lorsque vous souhaitez exécuter des tests sur des fichiers ou des répertoires spécifiques au sein de votre projet.

Usage

La `codebuild-glob-search` CLI utilise la syntaxe d'utilisation suivante :

```
codebuild-glob-search <glob_pattern1> [<glob_pattern2> ...]
```

- *<glob_pattern1><glob_pattern2>*, etc. : un ou plusieurs modèles globaux à comparer aux fichiers de votre répertoire de travail.
- ***: Correspond à n'importe quelle séquence de caractères (à l'exception des séparateurs de chemin).
- ****: Correspond à n'importe quelle séquence de caractères (y compris les séparateurs de chemin).

Note

Assurez-vous que la chaîne globale contient des guillemets. Pour vérifier les résultats de la correspondance de modèles, utilisez la `echo` commande.

```
version: 0.2

phases:
  build:
    commands:
      - echo $(codebuild-glob-search '**/__tests__/*.js')
      - codebuild-glob-search '**/__tests__/*.js' | xargs -n 1 echo
```

Output

La CLI produira une liste de chemins de fichiers séparés par de nouvelles lignes qui correspondent aux modèles globaux fournis. Les chemins de fichiers renvoyés seront relatifs au répertoire de travail.

Si aucun fichier correspondant aux modèles fournis n'est trouvé, la CLI affiche un message indiquant qu'aucun fichier n'a été trouvé.

Notez que les répertoires trouvés selon un modèle donné seront exclus des résultats de recherche.

Exemple

Si vous souhaitez rechercher uniquement les fichiers du répertoire tests et de ses sous-répertoires dotés d'une `.js` extension, vous pouvez utiliser la commande suivante avec la `codebuild-glob-search` CLI :

```
codebuild-glob-search '**/__tests__/*.js'
```

Cette commande recherchera tous les fichiers portant une `.js` extension dans le `__tests__` répertoire et ses sous-répertoires, comme indiqué par le modèle.

À propos du fractionnement des tests

AWS CodeBuild la fonction de division des tests vous permet de paralléliser l'exécution de votre suite de tests sur plusieurs instances de calcul, réduisant ainsi la durée globale d'exécution des tests. Cette fonctionnalité est activée par le biais de la configuration par lots dans les paramètres de votre CodeBuild projet et de l'`codebuild-tests-run` utilitaire dans votre fichier `buildspec`.

Les tests sont divisés en fonction de la stratégie de partitionnement spécifiée. CodeBuild propose deux stratégies de partitionnement, comme indiqué ci-dessous :

Répartition égale

La stratégie de `equal-distribution` partitionnement divise les tests entre des versions parallèles en fonction de l'ordre alphabétique des noms de fichiers de test. Cette approche trie d'abord les fichiers de test, puis utilise une méthode basée sur des fragments pour les distribuer, en veillant à ce que les fichiers similaires soient regroupés pour les tests. Il est recommandé de le faire lorsqu'il s'agit d'un ensemble relativement restreint de fichiers de test. Bien que cette méthode vise à allouer un nombre approximativement égal de fichiers à chaque partition, avec une différence maximale d'un, elle ne garantit pas la stabilité. Lorsque des fichiers de test sont ajoutés ou supprimés dans les versions suivantes, la distribution des fichiers existants peut changer, ce qui peut entraîner une réaffectation entre les partitions.

Stabilité

La stratégie de `stability` partitionnement utilise un algorithme de hachage cohérent pour répartir les tests entre les partitions, garantissant ainsi la stabilité de la distribution des fichiers. Lorsque de nouveaux fichiers sont ajoutés ou supprimés, cette approche garantit que les `file-to-shard` assignations existantes restent largement inchangées. Pour les suites de tests de grande taille, il est recommandé d'utiliser l'option de stabilité pour répartir uniformément les tests sur les partitions. Ce mécanisme vise à fournir une distribution quasi égale, en veillant à ce que chaque partition reçoive un nombre similaire de fichiers, avec une variance minimale. Bien que la stratégie de stabilité ne garantisse pas une distribution égale idéale, elle propose une distribution quasi égale qui permet de maintenir la cohérence des attributions de fichiers entre les versions, même lorsque des fichiers sont ajoutés ou supprimés.

Pour activer le fractionnement des tests, vous devez configurer la section par lots dans les paramètres de votre CodeBuild projet, en spécifiant le `parallelism` niveau souhaité et les autres paramètres pertinents. En outre, vous devrez inclure `codebuild-tests-run` utilitaire dans votre fichier `buildspec`, ainsi que les commandes de test et la méthode de division appropriées.

Fusionnez automatiquement les rapports de construction individuels

Dans les versions par lots Fanout, AWS CodeBuild prend en charge la fusion automatique des rapports de construction individuels en un rapport consolidé par lots. Cette fonctionnalité fournit une vue complète des résultats des tests et de la couverture du code pour toutes les versions d'un lot.

Comment ça marche

Lors de l'exécution de builds fanout par lots, chaque build génère [des rapports de test](#). CodeBuild consolide ensuite automatiquement les rapports identiques provenant de différentes versions dans un rapport unifié, qui est joint à la génération par lots. Ces rapports consolidés sont facilement accessibles via le `reportArns` champ de l' [BatchGetBuildBatches](#) API et peuvent également être consultés dans l'onglet Rapports de la console. Cette fonctionnalité de fusion s'étend également aux rapports découverts automatiquement.

Les rapports consolidés sont créés sous [des groupes de rapports](#) qui sont soit spécifiés dans le `buildspec`, soit découverts automatiquement par CodeBuild. Vous pouvez analyser les tendances des rapports fusionnés directement sous ces groupes de rapports, fournissant ainsi des informations précieuses sur les performances globales de construction et les indicateurs de qualité des versions historiques d'un même projet de génération par lots.

Pour chaque build individuel du lot, des groupes de rapports distincts CodeBuild sont automatiquement créés. Ils suivent une convention de dénomination spécifique, combinant le nom du groupe de rapports de génération par lots avec le suffixe `BuildFanoutShard<shard_number>`, où le `shard_number` représente le numéro de la partition dans laquelle le groupe de rapports est créé. Cette organisation vous permet de suivre et d'analyser les tendances au niveau des versions consolidées et individuelles, offrant ainsi une flexibilité dans la manière dont vous surveillez et évaluez leurs processus de création.

Le rapport de construction par lots suit la même structure que les rapports de [génération individuels](#). Les champs clés suivants de l'onglet Rapport sont spécifiques aux rapports de génération par lots :

État du rapport de génération par lots

L'état des rapports de génération par lots suit des règles spécifiques en fonction du type de rapport :

- Rapports d'essais :
 - Réussite : le statut est défini sur Réussite lorsque tous les rapports de construction individuels ont abouti.
 - Échec : le statut est défini sur Échec si un rapport de génération individuel a échoué.
 - Incomplet : le statut est marqué comme incomplet si un rapport de construction individuel est manquant ou a un statut incomplet.
- Rapports sur la couverture du code :
 - Terminé : le statut est défini comme terminé lorsque tous les rapports de construction individuels sont terminés.
 - Échec : le statut est défini sur Échec si un rapport de génération individuel a échoué.
 - Incomplet : le statut est marqué comme incomplet si un rapport de construction individuel est manquant ou a un statut incomplet.

Résumé du test

Le rapport de test fusionné consolide les champs suivants de tous les rapports de construction individuels :

- duration-in-nano-seconds: durée maximale du test en nanosecondes pour tous les rapports de construction individuels.
- total : le nombre combiné de tous les cas de test, en additionnant le nombre total de tests de chaque version.
- status-counts : fournit une vue consolidée des statuts de test tels que réussis, échoués ou ignorés, calculée en agrégeant le nombre de chaque type de statut sur toutes les versions individuelles.

Résumé de la couverture du code

Le rapport de couverture du code fusionné combine les champs de toutes les versions individuelles en utilisant les calculs suivants :

- branches couvertes : somme de toutes les branches couvertes à partir des rapports individuels.
- branches-missed : somme de toutes les branches manquantes dans les rapports individuels.
- branch-coverage-percentage: $(\text{Total covered branches} / \text{Total branches}) * 100$

- lignes couvertes : somme de toutes les lignes couvertes dans les rapports individuels.
- lines-missed : somme de toutes les lignes manquées dans les rapports individuels.
- lines-coverage-percentage: $(\text{Total covered lines} / \text{Total lines}) * 100$

ID d'exécution

L'ARN de génération par lots.

Cas de test

Le rapport fusionné contient une liste consolidée de tous les cas de test issus de versions individuelles, accessible via l'[DescribeTestCases](#) API et le rapport de génération par lots dans la console.

Couvertures par code

Le rapport de couverture du code fusionné fournit des informations consolidées sur la couverture des lignes et des branches pour chaque fichier dans toutes les versions individuelles, accessibles via l'[DescribeCodeCoverages](#) API et le rapport de génération par lots dans la console. Remarque : Pour les fichiers couverts par plusieurs fichiers de test répartis sur différentes partitions, le rapport fusionné utilise les critères de sélection suivants :

1. La sélection principale est basée sur la couverture de lignes la plus élevée parmi les fragments.
2. Si la couverture des lignes est égale sur plusieurs partitions, la partition présentant la couverture de branches la plus élevée est sélectionnée.

Exemple d'exécution de tests en parallèle pour différents frameworks de test

Vous pouvez utiliser la commande `codebuild-tests-run` CLI pour diviser et exécuter vos tests dans des environnements d'exécution parallèles. La section suivante fournit `buildspec.yml` des exemples de différents frameworks, illustrant l'utilisation de la `codebuild-tests-run` commande.

- Chaque exemple ci-dessous inclut un `parallelism` niveau de cinq, ce qui signifie que cinq environnements d'exécution identiques seront créés pour répartir vos tests. Vous pouvez choisir un `parallelism` niveau adapté à votre projet en modifiant la `parallelism` valeur dans la `build-fanout` section.
- Chaque exemple ci-dessous montre comment configurer vos tests pour qu'ils soient divisés par le nom du fichier de test, qui est par défaut. Cela répartit les tests de manière uniforme dans les environnements d'exécution parallèle.

Avant de commencer, consultez [Exécuter des tests parallèles dans des builds par lots](#) pour plus d'informations.

Pour obtenir la liste complète des options disponibles lors de l'utilisation de la commande `codebuild-tests-run` CLI, consultez [Utilisez la `codebuild-tests-run` commande CLI](#).

Rubriques

- [Configurer des tests parallèles avec Django](#)
- [Configurer des tests parallèles avec Elixir](#)
- [Configurer des tests parallèles avec Go](#)
- [Configuration de tests parallèles avec Java \(Maven\)](#)
- [Configurer des tests parallèles avec Javascript \(Jest\)](#)
- [Configurer des tests parallèles avec Kotlin](#)
- [Configurez des tests parallèles avec PHPUnit](#)
- [Configurer des tests parallèles avec Pytest](#)
- [Configurer des tests parallèles avec Ruby \(Cucumber\)](#)
- [Configurer des tests parallèles avec Ruby \(RSpec\)](#)

Configurer des tests parallèles avec Django

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Django sur une plate-forme Ubuntu :

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - sudo yum install -y python3 python3-pip
      - python3 -m ensurepip --upgrade
      - python3 -m pip install django
```

```

pre_build:
  commands:
    - echo 'Prebuild'
build:
  commands:
    - echo 'Running Django Tests'
    - |
      codebuild-tests-run \
        --test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES"
| sed -E "s/\//__/g; s/\.py$/; s/__/./g")' \
        --files-search "codebuild-glob-search '**/tests/*test_*.py'" \
        --sharding-strategy 'equal-distribution'
post_build:
  commands:
    - echo 'Test execution completed'

```

L'exemple ci-dessus montre l'utilisation de la variable d'environnement `CODEBUILD_CURRENT_SHARD_FILES`. Ici `CODEBUILD_CURRENT_SHARD_FILES` est utilisé pour récupérer les chemins des fichiers de notation par points supportés par Django. Utilisez des guillemets doubles `CODEBUILD_CURRENT_SHARD_FILES` à l'intérieur, comme indiqué ci-dessus.

Configurer des tests parallèles avec Elixir

Voici un exemple de `test buildspect.yml` illustrant l'exécution parallèle d'un test avec Elixir sur une plateforme Ubuntu :

```

version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Elixir dependencies'
      - sudo apt update
      - sudo DEBIAN_FRONTEND=noninteractive apt install -y elixir
      - elixir --version
      - mix --version

```

```
pre_build:
  commands:
    - echo 'Prebuild'
build:
  commands:
    - echo 'Running Elixir Tests'
    - |
      codebuild-tests-run \
        --test-command 'mix test' \
        --files-search "codebuild-glob-search '**/test/**/*_test.exs'" \
        --sharding-strategy 'equal-distribution'
post_build:
  commands:
    - echo "Test execution completed"
```

Configurer des tests parallèles avec Go

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Go sur une plateforme Linux :

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Fetching Go version'
      - go version
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running go Tests'
      - go mod init calculator
      - cd calc
      - |
        codebuild-tests-run \
```

```

--test-command "go test -v calculator.go" \
--files-search "codebuild-glob-search '**/*test.go'"
post_build:
  commands:
    - echo "Test execution completed"

```

Dans l'exemple ci-dessus, `calculator.go` la fonction contient des fonctions mathématiques simples à tester et tous les fichiers de test se trouvent `calc` dans un dossier. `calculator.go`

Configuration de tests parallèles avec Java (Maven)

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Java sur une plate-forme Linux :

```

version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo "Running mvn test"
      - |
        codebuild-tests-run \
          --test-command 'mvn test -Dtest=$(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
"s|src/test/java/||g; s|\.java//g; s|/|.|g; s/ /,/g" | tr "\n" "," | sed "s/,,$//")' \
          --files-search "codebuild-glob-search '**/test/**/*.*java'"

  post_build:
    commands:
      - echo "Running post-build steps..."
      - echo "Test execution completed"

```

Dans l'exemple donné, la variable d'environnement `CODEBUILD_CURRENT_SHARD_FILES` contient des fichiers de test dans le shard actuel, séparés par des nouvelles lignes. Ces fichiers sont

convertis en une liste de noms de classes séparés par des virgules au format accepté par le `-Dtest` paramètre pour Maven.

Configurer des tests parallèles avec Javascript (Jest)

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Javascript sur une plateforme Ubuntu :

```
version: 0.2

batch:
  fast-fail: true
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Node.js dependencies'
      - apt-get update
      - apt-get install -y nodejs
      - npm install
      - npm install --save-dev jest-junit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running JavaScript Tests'
      - |
        codebuild-tests-run \
          --test-command "npx jest" \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'stability'
  post_build:
    commands:
      - echo 'Test execution completed'
```

Configurer des tests parallèles avec Kotlin

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Kotlin sur une plate-forme Linux :

```

version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 2
    ignore-failure: false

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - echo 'Installing dependencies'
      - KOTLIN_VERSION="1.8.20" # Replace with your desired version
      - curl -o kotlin-compiler.zip -L "https://github.com/JetBrains/kotlin/releases/
download/v${KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip"
      - unzip kotlin-compiler.zip -d /usr/local
      - export PATH=$PATH:/usr/local/kotlinc/bin
      - kotlin -version
      - curl -O https://repo1.maven.org/maven2/org/junit/platform/junit-platform-
console-standalone/1.8.2/junit-platform-console-standalone-1.8.2.jar
    pre_build:
      commands:
        - echo 'prebuild'
    build:
      commands:
        - echo 'Running Kotlin Tests'
        - |
          codebuild-tests-run \
            --test-command 'kotlinc src/main/kotlin/*.kt $(echo
"$CODEBUILD_CURRENT_SHARD_FILES" | tr "\n" " ") -d classes -cp junit-platform-console-
standalone-1.8.2.jar' \
            --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
        - |
          codebuild-tests-run \
            --test-command '
          java -jar junit-platform-console-standalone-1.8.2.jar --class-path classes
\
          $(for file in $CODEBUILD_CURRENT_SHARD_FILES; do
            class_name=$(basename "$file" .kt)
            echo "--select-class $class_name"
          done)

```



```
    ' \
    --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
post_build:
  commands:
    - echo "Test execution completed"
```

Dans l'exemple ci-dessus, la `codebuild-tests-run` CLI est utilisée deux fois. Lors de la première exécution, `kotlinc` compile les fichiers. La `CODEBUILD_CURRENT_SHARD_FILES` variable récupère les fichiers de test affectés à la partition actuelle, qui sont ensuite convertis en une liste séparée par des espaces. Lors de la deuxième exécution, JUnit exécute les tests. Encore une fois, `CODEBUILD_CURRENT_SHARD_FILES` récupère les fichiers de test assignés à la partition actuelle, mais cette fois ils sont convertis en noms de classes.

Configurez des tests parallèles avec PHPUnit

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle PHPUnit sur une plateforme Linux :

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Install dependencies'
      - composer require --dev phpunit/phpunit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running phpunit Tests'
      - composer dump-autoload
      - |
        codebuild-tests-run \
          --test-command "./vendor/bin/phpunit --debug" \
          --files-search "codebuild-glob-search '**/tests/*Test.php'"

```

```
post_build:
  commands:
    - echo 'Test execution completed'
```

Configurer des tests parallèles avec Pytest

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Pytest sur une plate-forme Ubuntu :

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - apt-get update
      - apt-get install -y python3 python3-pip
      - pip3 install --upgrade pip
      - pip3 install pytest
  build:
    commands:
      - echo 'Running Python Tests'
      - |
        codebuild-tests-run \
          --test-command 'python -m pytest' \
          --files-search "codebuild-glob-search 'tests/test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle avec Pytest sur une plate-forme Windows :

```
version: 0.2
```

```

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - pip install pytest
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running pytest'
      - |
        & codebuild-tests-run `
        --test-command 'pytest @("$env:CODEBUILD_CURRENT_SHARD_FILES" -split \"`r?`n
        \")' `
        --files-search "codebuild-glob-search '**/test_*.py' '**/*_test.py'" `
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"

```

Dans l'exemple ci-dessus, la variable d'`CODEBUILD_CURRENT_SHARD_FILE` environnement est utilisée pour récupérer les fichiers de test affectés à la partition actuelle et transmis sous forme de tableau à la commande `pytest`.

Configurer des tests parallèles avec Ruby (Cucumber)

Voici un `buildspec.yml` exemple illustrant l'exécution de tests parallèles avec Cucumber sur une plate-forme Linux :

```

version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

```

```
phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - gem install bundler
      - bundle install
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Cucumber Tests'
      - cucumber --init
      - |
        codebuild-tests-run \
          --test-command "cucumber" \
          --files-search "codebuild-glob-search '**/*.feature'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Configurer des tests parallèles avec Ruby (RSpec)

Voici un `buildspec.yml` exemple illustrant l'exécution de tests en parallèle RSpec sur une plateforme Ubuntu :

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - apt-get update
      - apt-get install -y ruby ruby-dev build-essential
      - gem install bundler
      - bundle install
```

```
build:
  commands:
    - echo 'Running Ruby Tests'
    - |
      codebuild-tests-run \
        --test-command 'bundle exec rspec' \
        --files-search "codebuild-glob-search 'spec/**/*.rb'" \
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Des mises en cache pour améliorer les performances

Vous pouvez faire des économies lors de vos générations de projet en utilisant un cache. Un cache peut stocker les éléments réutilisables de votre environnement de génération et les utiliser sur plusieurs générations. Votre projet de génération peut utiliser l'un des deux types de mise en cache suivants : Amazon S3 ou local. Si vous utilisez un cache local, vous devez choisir un ou plusieurs des trois modes de cache : le cache source, le cache de couche Docker et le cache personnalisé.

Note

Le mode de cache de couche Docker est disponible pour l'environnement Linux uniquement. Si vous choisissez ce mode, vous devez exécuter votre build en mode privilégié. CodeBuild le mode privilégié des projets accorde à son conteneur l'accès à tous les appareils. Pour de plus amples informations, veuillez consulter [Privilège d'exécution et fonctionnalités Linux](#) sur le site web Docker Docs.

Rubriques

- [Mise en cache d'Amazon S3](#)
- [Mise en cache locale](#)
- [Spécifier un cache local](#)

Mise en cache d'Amazon S3

La mise en cache Amazon S3 stocke le cache dans un compartiment Amazon S3 disponible sur plusieurs hôtes de build. C'est une bonne option pour les artefacts de construction de petite ou moyenne taille qui sont plus coûteux à créer qu'à télécharger.

Pour utiliser Amazon S3 dans une version, vous pouvez spécifier les chemins des fichiers que vous souhaitez mettre en cache dans `buildspec.yml`. CodeBuild stockera et mettra automatiquement à jour le cache à l'emplacement Amazon S3 configuré sur le projet. Si vous ne spécifiez pas les chemins des fichiers, CodeBuild nous ferons de notre mieux pour mettre en cache les dépendances linguistiques courantes afin d'accélérer les compilations. Vous pouvez consulter les détails du cache dans les journaux de compilation.

En outre, si vous souhaitez disposer de plusieurs versions du cache, vous pouvez définir une clé de cache dans `buildspec.yml`. CodeBuild stocke le cache dans le contexte de cette clé de cache et crée une copie de cache unique qui ne sera pas mise à jour une fois créée. Les clés de cache peuvent également être partagées entre les projets. Les fonctionnalités telles que les clés dynamiques, le versionnement du cache et le partage du cache entre les versions ne sont disponibles que lorsqu'une clé est spécifiée.

Pour en savoir plus sur la syntaxe du cache dans le fichier `buildspec`, consultez [cache](#) la référence `buildspec`.

Rubriques

- [Génération de clés dynamiques](#)
- [codebuild-hash-files](#)
- [Version du cache](#)
- [Partage du cache entre projets](#)
- [Exemples de spécifications de construction](#)

Génération de clés dynamiques

Une clé de cache peut inclure des commandes shell et des variables d'environnement pour la rendre unique, ce qui permet des mises à jour automatiques du cache en cas de modification de clé. Par exemple, vous pouvez définir une clé en utilisant le hachage du `package-lock.json` fichier. Lorsque les dépendances de ce fichier changent, le hachage, et donc la clé de cache, change, ce qui déclenche la création automatique d'un nouveau cache.

```
cache:  
  key: npm-key-$(codebuild-hash-files package-lock.json)
```

CodeBuild évaluera l'expression `$(codebuild-hash-files package-lock.json)` pour obtenir la clé finale :

```
npm-key-abc123
```

Vous pouvez également définir une clé de cache à l'aide de variables d'environnement, telles que `CODEBUILD_RESOLVED_SOURCE_VERSION`. Cela garantit que chaque fois que votre source change, une nouvelle clé est générée, ce qui entraîne l'enregistrement automatique d'un nouveau cache :

```
cache:  
  key: npm-key-$CODEBUILD_RESOLVED_SOURCE_VERSION
```

CodeBuild évaluera l'expression et obtiendra la clé finale :

```
npm-key-046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

codebuild-hash-files

`codebuild-hash-files` est un outil CLI qui calcule un hachage SHA-256 pour un ensemble de fichiers du répertoire CodeBuild source à l'aide de modèles globaux :

```
codebuild-hash-files <glob-pattern-1> <glob-pattern-2> ...
```

Voici quelques exemples d'utilisation de `codebuild-hash-files` :

```
codebuild-hash-files package-lock.json  
codebuild-hash-files '**/*.md'
```

Version du cache

La version du cache est un hachage généré à partir des chemins des répertoires mis en cache. Si deux caches ont des versions différentes, ils sont traités comme des caches distincts pendant le processus de mise en correspondance. Par exemple, les deux caches suivants sont considérés comme différents car ils font référence à des chemins différents :

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "pip-dependencies/**/*"
```

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target tmp/pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "tmp/pip-dependencies/**/*"
```

Partage du cache entre projets

Vous pouvez utiliser le champ `cacheNamespace` API situé sous la `cache` section pour partager un cache entre plusieurs projets. Ce champ définit l'étendue du cache. Pour partager un cache, vous devez effectuer les opérations suivantes :

- Utilisez le même `cacheNamespace`.
- Spécifiez le même `cachekey`.
- Définissez des chemins de cache identiques.
- Utilisez les mêmes compartiments Amazon S3 et `pathPrefix` s'ils sont définis.

Cela garantit la cohérence et permet le partage du cache entre les projets.

Spécifier un espace de noms de cache (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Sélectionnez **Create a project** (Créer un projet). Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
3. Dans **Artefacts**, sélectionnez **Configuration supplémentaire**.
4. Pour le type de cache, choisissez **Amazon S3**.
5. Pour **Espace de noms du cache** - facultatif, entrez une valeur d'espace de noms.

▼ Additional configuration

Cache, encryption key

Encryption key - optional

Provide the AWS KMS customer master key used to encrypt this build's output artifacts. The default is your AWS-managed customer master key for S3.

arn:aws:kms:<region-ID>:<account-ID>:key/<key-ID>

Cache type

Amazon S3 ▼

Cache bucket**Cache path prefix - optional****Cache lifecycle (days) - optional**

You can apply a lifecycle expiration action to all or a subset of objects in the cache bucket based on the path prefix.

+ Add expiration**Cache namespace - optional**

test-cache-namespace

Provide a cache namespace if you want to share caches across projects.

6. Continuez avec les valeurs par défaut, puis choisissez **Create build project**.

Spécifiez un espace de noms de cache ()AWS CLI

Vous pouvez utiliser le `--cache` paramètre dans le AWS CLI pour spécifier un espace de noms de cache.

```
--cache '{"type": "S3", "location": "your-s3-bucket", "cacheNamespace": "test-cache-namespace"}'
```

Exemples de spécifications de construction

Voici plusieurs exemples de spécifications de construction pour les langages courants :

Rubriques

- [Cache les dépendances de Node.js](#)
- [Dépendances du cache en Python](#)
- [Dépendances du cache Ruby](#)
- [Dépendances de Cache Go](#)

Cache les dépendances de Node.js

Si votre projet inclut un `package-lock.json` fichier et l'utilise npm pour gérer les dépendances de Node.js, l'exemple suivant montre comment configurer la mise en cache. Par défaut, npm installe les dépendances dans le `node_modules` répertoire.

```
version: 0.2

phases:
  build:
    commands:
      - npm install
cache:
  key: npm-${codebuild-hash-files package-lock.json}
  paths:
    - "node_modules/**/*"
```

Dépendances du cache en Python

Si votre projet inclut un `requirements.txt` fichier et utilise pip pour gérer les dépendances Python, l'exemple suivant montre comment configurer la mise en cache. Par défaut, pip installe les packages dans le répertoire du `site-packages` système.

```
version: 0.2
```

```
phases:
  build:
    commands:
      - pip install -r requirements.txt
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "/root/.pyenv/versions/${python_version}/lib/python${python_major_version}/site-packages/**/*"
```

En outre, vous pouvez installer des dépendances dans un répertoire spécifique et configurer la mise en cache pour ce répertoire.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt --target python-dependencies
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "python-dependencies/**/*"
```

Dépendances du cache Ruby

Si votre projet inclut un `Gemfile.lock` fichier et l'utilise `Bundler` pour gérer les dépendances des gemmes, l'exemple suivant montre comment configurer efficacement la mise en cache.

```
version: 0.2

phases:
  build:
    commands:
      - bundle install --path vendor/bundle
cache:
  key: ruby-$(codebuild-hash-files Gemfile.lock)
  paths:
    - "vendor/bundle/**/*"
```

Dépendances de Cache Go

Si votre projet inclut un `go.sum` fichier et utilise des modules Go pour gérer les dépendances, l'exemple suivant montre comment configurer la mise en cache. Par défaut, les modules Go sont téléchargés et stockés dans le `${GOPATH}/pkg/mod` répertoire.

```
version: 0.2

phases:
  build:
    commands:
      - go mod download
cache:
  key: go-$(codebuild-hash-files go.sum)
  paths:
    - "/go/pkg/mod/**/*"
```

Mise en cache locale

La mise en cache locale stocke un cache localement sur un hôte de génération. Le cache est disponible uniquement sur cet hôte de génération. C'est une bonne option pour les artefacts de build de taille intermédiaire à grande, car le cache est immédiatement disponible sur l'hôte de build. Ce n'est pas la meilleure option si vos builds sont rares. Cela signifie que les performances de génération ne sont pas affectés par le temps de transfert réseau.

Si vous choisissez la mise en cache locale, vous devez choisir un ou plusieurs des modes de cache suivants :

- Le mode de cache source met en cache des métadonnées Git pour des sources principales et secondaires. Lorsque le cache est créé, les générations suivantes extraient uniquement les modifications apportées entre les validations. Ce mode est un bon choix pour les projets avec un répertoire de travail propre et une source correspondant à un référentiel Git de grande taille. Si vous choisissez cette option et que votre projet n'utilise pas de référentiel Git (AWS CodeCommit, GitHub, GitHub Enterprise Server ou Bitbucket), l'option est ignorée.
- Le mode de cache de couche Docker met en cache des couches Docker existantes. Ce mode est un bon choix pour les projets qui génèrent ou extraient des images Docker de taille importante. Il permet d'éviter les problèmes de performances causées par l'extraction d'images Docker de grande taille à partir du réseau.

 Note

- Vous pouvez utiliser un cache de couche Docker uniquement dans l'environnement Linux.
- L'indicateur `privileged` doit être défini de sorte que votre projet dispose des autorisations Docker requises.

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux Capabilities](#) sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

- Vous devez prendre en compte les implications de sécurité avant d'utiliser un cache de couche Docker.

- Le mode de cache personnalisé met en cache les répertoires que vous spécifiez dans le fichier `buildspec`. Ce mode est un bon choix si votre scénario de génération n'est pas adapté à un des deux autres modes de cache local. Si vous utilisez un cache personnalisé :
 - Seuls les répertoires peuvent être spécifiés pour la mise en cache. Vous ne pouvez pas spécifier de fichiers individuels.
 - Les liens symboliques sont utilisés pour référencer les répertoires mis en cache.
 - Les répertoires mis en cache sont liés à votre génération avant que ses sources de projet soient téléchargées. Les éléments mis en cache remplacent les éléments source s'ils portent le même nom. Les répertoires sont spécifiés à l'aide de chemins de cache dans le fichier `buildspec`. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).
 - Évitez les noms de répertoire identiques dans la source et dans le cache. Les répertoires mis en cache localement peuvent remplacer ou supprimer le contenu des répertoires du référentiel source portant le même nom.

 Note

La mise en cache locale n'est pas prise en charge avec le type `d'Linux_GPU_CONTAINER` environnement et le type de `BUILD_GENERAL1_2XLARGE`

calcul. Pour de plus amples informations, veuillez consulter [Modes et types de calcul de l'environnement de création](#).

Note

La mise en cache locale n'est pas prise en charge lorsque vous configurez CodeBuild pour fonctionner avec un VPC. Pour plus d'informations sur l'utilisation VPCs avec CodeBuild, consultez [Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Spécifier un cache local

Vous pouvez utiliser la console AWS CLI, le SDK ou CloudFormation pour spécifier un cache local. Pour plus d'informations sur la mise en cache locale, consultez [Mise en cache locale](#).

Rubriques

- [Spécification d'une mise en cache locale \(CLI\)](#)
- [Spécification d'une mise en cache locale \(console\)](#)
- [Spécification d'une mise en cache locale \(CloudFormation\)](#)

Spécification d'une mise en cache locale (CLI)

Vous pouvez utiliser le `--cache` paramètre dans le AWS CLI pour spécifier chacun des trois types de cache locaux.

- Pour spécifier un cache source :

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Pour spécifier un cache de couche Docker :

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

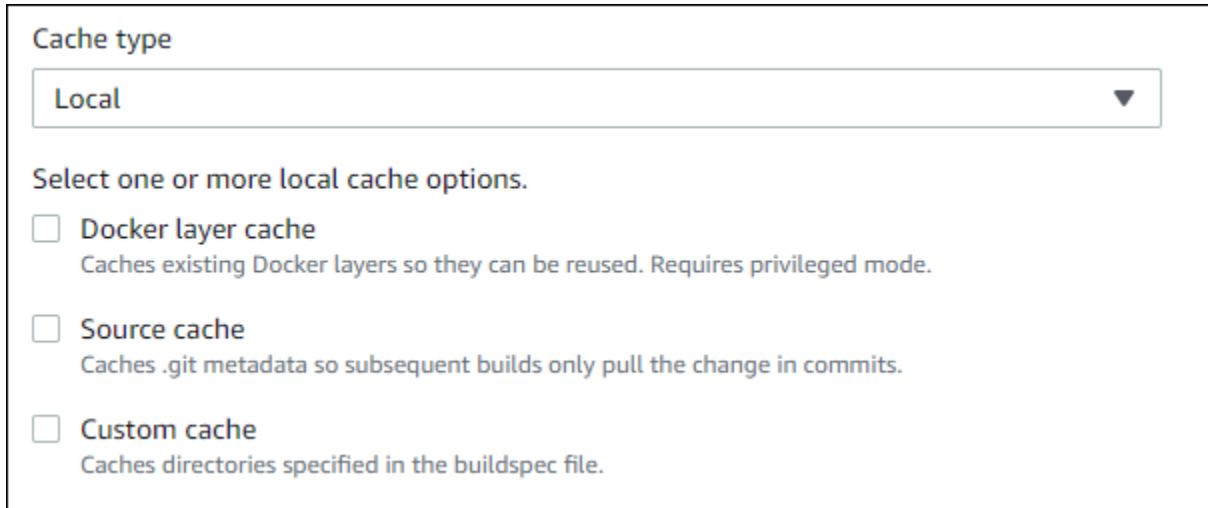
- Pour spécifier un cache personnalisé :

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

Spécification d'une mise en cache locale (console)

Vous spécifiez un cache dans la section Artifacts (Artefacts) de la console. Pour le type de cache, choisissez Amazon S3 ou Local. Si vous choisissez Local, choisissez une ou plusieurs des trois options de cache local.



The screenshot shows the 'Cache type' dropdown menu in the AWS CodeBuild console. The dropdown is set to 'Local'. Below the dropdown, there is a section titled 'Select one or more local cache options.' with three checkboxes and their descriptions:

- ☐ **Docker layer cache**
Caches existing Docker layers so they can be reused. Requires privileged mode.
- ☐ **Source cache**
Caches .git metadata so subsequent builds only pull the change in commits.
- ☐ **Custom cache**
Caches directories specified in the buildspec file.

Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).

Spécification d'une mise en cache locale (CloudFormation)

Si vous avez l'habitude de spécifier un cache local, sur la Cache propriété, pourType, spécifiezLOCAL. L'exemple de CloudFormation code au format YAML suivant spécifie les trois types de cache locaux. Vous pouvez spécifier n'importe quelle combinaison de types. Si vous utilisez un cache de couche Docker, sous Environment, vous devez définir PrivilegedMode sur true et Type sur LINUX_CONTAINER.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: <service-role>
    Artifacts:
      Type: S3
      Location: <bucket-name>
      Name: myArtifact
      EncryptionDisabled: true
      OverrideArtifactName: true
```

```
Environment:
  Type: LINUX_CONTAINER
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:5.0
  Certificate: <bucket/cert.zip>
  # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
  PrivilegedMode: true
Source:
  Type: GITHUB
  Location: <github-location>
  InsecureSsl: true
  GitCloneDepth: 1
  ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
  Type: LOCAL
  Modes: # You can specify one or more cache mode,
    - LOCAL_CUSTOM_CACHE
    - LOCAL_DOCKER_LAYER_CACHE
    - LOCAL_SOURCE_CACHE
```

Note

Par défaut, le démon Docker est activé pour les versions non VPC. Si vous souhaitez utiliser des conteneurs Docker pour les builds VPC, [consultez Runtime Privilege et Linux Capabilities](#) sur le site Web de Docker Docs et activez le mode privilégié. De plus, Windows ne prend pas en charge le mode privilégié.

Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(CloudFormation\)](#).

Debug s'intègre AWS CodeBuild

AWS CodeBuild fournit deux méthodes pour le débogage des versions pendant le développement et le dépannage. Vous pouvez utiliser l'environnement CodeBuild Sandbox pour étudier les problèmes et valider les correctifs en temps réel, ou vous pouvez utiliser le gestionnaire de session de AWS Systems Manager pour vous connecter au conteneur de construction et consulter l'état du conteneur.

Déboguer les builds avec Sandbox CodeBuild

L'environnement CodeBuild sandbox fournit une session de débogage interactive dans un environnement sécurisé et isolé. Vous pouvez interagir avec l'environnement directement via le AWS Management Console ou AWS CLI, exécuter des commandes et valider votre processus de création étape par étape. Il utilise un modèle de facturation rentable à la seconde et prend en charge la même intégration native avec les fournisseurs de sources et de AWS services que votre environnement de construction. Vous pouvez également vous connecter à un environnement sandbox à l'aide de clients SSH ou à partir de vos environnements de développement intégrés (IDEs).

Pour en savoir plus sur la tarification du CodeBuild sandbox, consultez la [documentation CodeBuild tarifaire](#). Pour obtenir des instructions détaillées, consultez la [Déboguer les builds avec Sandbox CodeBuild](#) documentation.

Déboguer les builds avec le gestionnaire de session

AWS Systems Manager Session Manager permet d'accéder directement aux versions en cours d'exécution dans leur environnement d'exécution réel. Cette approche vous permet de vous connecter à des conteneurs de build actifs et d'inspecter le processus de construction en temps réel. Vous pouvez examiner le système de fichiers, surveiller les processus en cours d'exécution et résoudre les problèmes au fur et à mesure qu'ils surviennent.

Pour obtenir des instructions détaillées, consultez la [Déboguer les builds avec le gestionnaire de session](#) documentation.

Déboguer les builds avec Sandbox CodeBuild

Dans AWS CodeBuild, vous pouvez déboguer une version en utilisant le CodeBuild sandbox pour exécuter des commandes personnalisées et résoudre les problèmes liés à votre version.

Rubriques

- [Conditions préalables](#)
- [Déboguer les versions avec CodeBuild sandbox \(console\)](#)
- [Déboguer les builds avec CodeBuild sandbox \(AWS CLI\)](#)
- [Tutoriel : Connexion à un sandbox via SSH](#)
- [Résolution des problèmes AWS CodeBuild de connexion SSH au sandbox](#)

Conditions préalables

Avant d'utiliser un CodeBuild sandbox, assurez-vous que votre rôle de CodeBuild service est conforme à la politique SSM suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:codebuild:us-east-1:111122223333:build/*",
        "arn:aws:ssm:us-east-1::document/AWS-StartSSHSession"
      ]
    }
  ]
}
```

Déboguer les versions avec CodeBuild sandbox (console)

Suivez les instructions suivantes pour exécuter des commandes et connecter votre client SSH au CodeBuild sandbox de la console.

Exécuter des commandes avec le CodeBuild bac à sable (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération. Choisissez le projet de génération, puis choisissez Debug build.

The screenshot shows the AWS CodeBuild console for a project named 'sandbox-project'. At the top, there are buttons for 'Actions', 'Create trigger', 'Edit', 'Clone', 'Clear cache', 'Debug build', 'Start build with overrides', and 'Start build'. Below this is the 'Configuration' section with a table:

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::[redacted]:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

Below the table are tabs for 'Build history', 'Batch history', 'Project details' (selected), 'Build triggers', 'Metrics', and 'Debug sessions'. The 'Project configuration' section shows:

Name	Description
sandbox-project	-
Project ARN	Build badge
arn:aws:codebuild:us-east-1:[redacted]:project/sandbox-project	Disabled

An 'Edit' button is visible in the top right of the 'Project configuration' section.

3. Dans l'onglet Exécuter la commande, entrez vos commandes personnalisées, puis choisissez Exécuter la commande.

The screenshot shows the 'Debug build' page in the AWS CodeBuild console. At the top, there are tabs for 'Run Command' (selected), 'SSH Client', and 'Session Manager'. Below this is a blue information box titled 'Run custom commands with sandbox' with a 'Learn more' link. It contains the following bullet points:

- Launches a sandbox environment mirroring your project configuration.
- Automatically downloads source code, while skipping project buildspec execution.
- Ideal for reproducing failure, experimenting fixes and investigation.

Below the information box is a 'Command' input area with a text box containing '1 pwd'. To the right of the text box are icons for 'Run' and 'Cancel'. At the bottom left of the command area are icons for 'Copy' and 'Paste'. At the bottom right is a status bar showing '1:4 SH'. A 'Run command' button is located at the bottom of the page.

4. Votre CodeBuild sandbox sera ensuite initialisé et commencera à exécuter vos commandes personnalisées. La sortie sera affichée dans l'onglet Sortie lorsqu'elle sera terminée.

Debug build

Run Command | SSH Client | Session Manager

✓ **Sandbox is running**
Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is ready and available for use. Stop sandbox

Command

```
1 pwd
```

⊞ ⊞ 0 0 1:1 SH

Run command

Command output | Sandbox phases | Sandbox logs | Sandbox configurations | Command history

View entire log in [CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

5. Lorsque le dépannage est terminé, vous pouvez arrêter le bac à sable en choisissant Arrêter le bac à sable. Choisissez ensuite Stop pour confirmer que votre sandbox sera arrêté.

Stop sandbox ✕

Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

- `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18`

Cancel **Stop**

Debug build

[Run Command](#)[SSH Client](#)[Session Manager](#)**Sandbox is stopped**Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is currently inactive.[Command output](#)[Sandbox phases](#)[Sandbox logs](#)[Sandbox configurations](#)[Command history](#)[View entire log in CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

Connectez-vous à votre client SSH avec CodeBuild sandbox (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération. Choisissez le projet de génération, puis choisissez Debug build.

sandbox-project [Actions ▼](#) [Create trigger](#) [Edit](#) [Clone](#) [Clear cache](#) [Debug build](#) [Start build with overrides](#) [Start build](#)

Configuration

Source provider No source	Primary repository -	Artifacts upload location -	Service role <code>arn:aws:iam::[redacted]:role/service-role/codebuild-sandbox-project-service-role</code>
Public builds Disabled			

[Build history](#) [Batch history](#) [Project details](#) [Build triggers](#) [Metrics](#) [Debug sessions](#)

Project configuration [Edit](#)

Name sandbox-project	Description -
Project ARN ☐ <code>arn:aws:codebuild:us-east-1:[redacted]:project/sandbox-project</code>	Build badge Disabled

3. Dans l'onglet Client SSH, choisissez Start sandbox.

Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build

Debug build

Run Command | **SSH Client** | Session Manager

Connect to your SSH client with sandbox

- Launches a sandbox environment with SSH connectivity.
- Connect directly using SSH clients or your preferred IDE.

[Learn more](#)

Start sandbox

4. Une fois le CodeBuild sandbox lancé, suivez les instructions de la console pour connecter votre client SSH au sandbox.

Debug build

Run Command | **SSH Client** | Session Manager

Sandbox is running

Your sandbox `sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8` is ready and available for use.

Stop sandbox

Terminal | Visual Studio Code | IntelliJ IDEA

Linux | **macOS** | Windows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Make sure your CLI user has the `codebuild:StartSandboxConnection` permission. For more information, see [AWS CLI authentication](#) documentation.

Connect to your sandbox environment with following command:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:██████████:sandbox/sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8
```

5. Lorsque le dépannage est terminé, vous pouvez arrêter le bac à sable en choisissant Arrêter le bac à sable. Choisissez ensuite Stop pour confirmer que votre sandbox sera arrêté.

Stop sandbox

×

Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

- sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8

Cancel

Stop

Debug build

Run Command

SSH Client

Session Manager

⚠

Sandbox is stopped

Your sandbox `sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8` is currently inactive.

Sandbox phases

Sandbox logs

Sandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
QUEUED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
PROVISIONING	✔ Succeeded		5 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	✔ Succeeded		8 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
RUN_SANDBOX	✔ Succeeded		213 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
UPLOAD_ARTIFACTS	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
FINALIZING	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
COMPLETED	✔ Succeeded	-	-	Apr 8, 2025 1:36 PM (UTC-7:00)	-

Déboguer les builds avec CodeBuild sandbox ()AWS CLI

Suivez les instructions suivantes pour exécuter des commandes et connecter votre client SSH au CodeBuild sandbox.

Démarrer un CodeBuild bac à sable ()AWS CLI

CLI command

```
aws codebuild start-sandbox --project-name $PROJECT_NAME
```

- `--project-name`: nom CodeBuild du projet

Sample request

```
aws codebuild start-sandbox --project-name "project-name"
```

Sample response

```
{
  "id": "project-name",
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
  "projectName": "project-name",
  "requestTime": "2025-02-06T11:24:15.560000-08:00",
  "status": "QUEUED",
  "source": {
    "type": "S3",
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-sources/eb-sample-jetty-v4.zip",
    "insecureSsl": false
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:6.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [{
      "name": "foo",
      "value": "bar",
      "type": "PLAINTEXT"
    },
    {
      "name": "bar",
      "value": "baz",
      "type": "PLAINTEXT"
    }
  ],
  "privilegedMode": false,
  "imagePullCredentialsType": "CODEBUILD"
},
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
```



```

        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "QUEUED",
    "status": "QUEUED",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "logs": {
        "groupName": "group",
        "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
            "encryptionDisabled": false
        }
    }
}
}

```

```
}
```

Obtenir des informations sur l'état du bac à sable (AWS CLI)

CLI command

```
aws codebuild batch-get-sandboxes --ids $SANDBOX_IDS
```

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

- `--ids`: liste séparée par des virgules de `sandboxIds` ou `sandboxArns`.

Vous pouvez fournir un identifiant de sandbox ou un ARN de sandbox :

- Identifiant du bac à sable : `<codebuild-project-name>: <UUID>`

Par exemple, `project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

- ARN du bac à sable : `arn:aws:codebuild : ::sandbox/ : <region> <account-id>
<codebuild-project-name> <UUID>`

Par exemple, `arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

Sample response

```
{  
  "sandboxes": [{  
    "id": "project-name",  
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
    "projectName": "project-name",  
    "requestTime": "2025-02-06T11:24:15.560000-08:00",  
    "endTime": "2025-02-06T11:39:21.587000-08:00",  
    "status": "STOPPED",  
    "source": {  
      "type": "S3",
```

```

        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
            "name": "foo",
            "value": "bar",
            "type": "PLAINTEXT"
        },
        {
            "name": "bar",
            "value": "baz",
            "type": "PLAINTEXT"
        }
    ],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"timeoutInMinutes": 10,
"queuedTimeoutInMinutes": 480,
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/
SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "COMPLETED",
    "status": "STOPPED",

```

```

"startTime": "2025-02-06T11:24:15.626000-08:00",
"endTime": "2025-02-06T11:39:21.600000-08:00",
"phases": [{
  "phaseType": "SUBMITTED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.577000-08:00",
  "endTime": "2025-02-06T11:24:15.606000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "QUEUED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.606000-08:00",
  "endTime": "2025-02-06T11:24:16.067000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "PROVISIONING",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:16.067000-08:00",
  "endTime": "2025-02-06T11:24:20.519000-08:00",
  "durationInSeconds": 4,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "DOWNLOAD_SOURCE",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:20.519000-08:00",
  "endTime": "2025-02-06T11:24:22.238000-08:00",
  "durationInSeconds": 1,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "RUNNING_SANDBOX",
  "phaseStatus": "TIMED_OUT",
  "startTime": "2025-02-06T11:24:22.238000-08:00",
  "endTime": "2025-02-06T11:39:21.560000-08:00",
  "durationInSeconds": 899,

```

```

        "contexts": [{
            "statusCode": "BUILD_TIMED_OUT",
            "message": "Build has timed out. "
        }]
    },
    {
        "phaseType": "COMPLETED",
        "startTime": "2025-02-06T11:39:21.560000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}
}],
"sandboxesNotFound": []
}

```

Arrêter un bac à sable ()AWS CLI

CLI command

```
aws codebuild stop-sandbox --id $SANDBOX-ID
```

- `--id`: A sandboxId ou sandboxArn.

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

Sample response

```
{  
  "id": "project-name",  
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
  "projectName": "project-name",  
  "requestTime": "2025-02-06T11:24:15.560000-08:00",  
  "status": "STOPPING",  
  "source": {  
    "type": "S3",  
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-  
sources/eb-sample-jetty-v4.zip",  
    "insecureSsl": false  
  },  
  "environment": {  
    "type": "LINUX_CONTAINER",  
    "image": "aws/codebuild/standard:6.0",  
    "computeType": "BUILD_GENERAL1_SMALL",  
    "environmentVariables": [{  
      "name": "foo",  
      "value": "bar",  
      "type": "PLAINTEXT"  
    },  
    {  
      "name": "bar",  
      "value": "baz",  
      "type": "PLAINTEXT"  
    }  
  ],  
  ],  
}
```

```

    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
  },
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
      "status": "ENABLED",
      "groupName": "group",
      "streamName": "stream"
    },
    "s3Logs": {
      "status": "ENABLED",
      "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
      "encryptionDisabled": false
    }
  },
  "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
  "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
  "currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "RUN_SANDBOX",
    "status": "STOPPING",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "phases": [{
      "phaseType": "SUBMITTED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.144000-08:00",
      "endTime": "2025-02-08T14:33:26.173000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "QUEUED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.173000-08:00",
      "endTime": "2025-02-08T14:33:26.702000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "PROVISIONING",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.702000-08:00",
      "endTime": "2025-02-08T14:33:30.530000-08:00",
      "durationInSeconds": 3,

```

```

        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "DOWNLOAD_SOURCE",
        "phaseStatus": "SUCCEEDED",
        "startTime": "2025-02-08T14:33:30.530000-08:00",
        "endTime": "2025-02-08T14:33:33.478000-08:00",
        "durationInSeconds": 2,
        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "RUN_SANDBOX",
        "startTime": "2025-02-08T14:33:33.478000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
}

```



```
    }  
  }  
}
```

Lancer l'exécution d'une commande (AWS CLI)

CLI command

```
aws codebuild start-command-execution --command $COMMAND --type $TYPE --sandbox-id  
$SANDBOX-ID
```

- `--command`: commande qui doit être exécutée.
- `--sandbox-id`: A sandboxId ou sandboxArn.
- `--type`: le type de commande, SHELL.

Sample request

```
aws codebuild start-command-execution --command "echo \"Hello World\"" --type SHELL --  
sandbox-id "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{  
  "id": "e1c658c2-02bb-42a8-9abb-94835241fcd6",  
  "sandboxId": "f7126a4a-b0d5-452f-814c-fea73718f805",  
  "submitTime": "2025-02-06T20:12:02.683000-08:00",  
  "status": "SUBMITTED",  
  "command": "echo \"Hello World\"",  
  "type": "SHELL",  
  "logs": {  
    "groupName": "group",  
    "streamName": "stream",  
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-  
west-2#logsV2:log-groups/log-group/group/log-events/stream",  
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-  
pool-1-us-west-2-beta-default-build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz?  
region=us-west-2",  
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-  
group:group:log-stream:stream",  
  }  
}
```

```

    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-
build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}

```

Obtenir des informations sur les exécutions de commandes (AWS CLI)

CLI command

```
aws codebuild batch-get-command-executions --command-execution-ids $COMMAND-IDS --
sandbox-id $SANDBOX-IDS
```

- `--command-execution-ids`: liste séparée par des virgules de `commandExecutionIds`
- `--sandbox-id`: A `sandboxId` ou `sandboxArn`.

Sample request

```
aws codebuild batch-get-command-executions --command-execution-
ids "c3c085ed-5a8f-4531-8e95-87d547f27ffd" --sandbox-id "arn:aws:codebuild:us-
west-2:962803963624:sandbox/project-name"
```

Sample response

```

{
  "commandExecutions": [{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",

```

```

    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
      }
    },
    "commandExecutionsNotFound": []
  }
}

```

Répertorier les exécutions de commandes pour un sandbox ()AWS CLI

CLI command

```
aws codebuild list-command-executions-for-sandbox --sandbox-id $SANDBOX-ID --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --sort-order $SORT_ORDER
```

- `--next-token`: le jeton suivant, le cas échéant, pour obtenir des résultats paginés. Vous obtiendrez cette valeur lors de l'exécution précédente de listes de sandbox.

- `--max-results`: (Facultatif) Le nombre maximum d'enregistrements sandbox à récupérer.
- `--sort-order`: ordre dans lequel les enregistrements sandbox doivent être récupérés.

Sample request

```
aws codebuild list-command-executions-for-sandbox --sandbox-id
"arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
  "commandExecutions": [{
    "id": "aad6687e-07bc-45ab-a1fd-f5440229b528",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:35.304000-08:00",
    "startTime": "2025-02-10T20:18:35.615000-08:00",
    "endTime": "2025-02-10T20:18:35.651000-08:00",
    "status": "FAILED",
    "command": "fail command",
    "type": "SHELL",
    "exitCode": "127",
    "standardErrContent": "/codebuild/output/tmp/script.sh: 4: fail: not
found\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
```

```

        "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
        "encryptionDisabled": false
    }
},
{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",
    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
        "groupName": "group",
        "streamName": "stream",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
            "encryptionDisabled": false
        }
    }
}
]

```

```
}
```

Répertorier les bacs à sable ()AWS CLI

CLI command

```
aws codebuild list-sandboxes --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --  
sort-order $SORT_ORDER
```

Sample request

```
aws codebuild list-sandboxes
```

Sample response

```
{  
  "ids": [  
    "s3-log-project-integ-test-temp173925062814985d64e0f-7880-41df-9a3c-  
fb6597a266d2:827a5243-0841-4b69-a720-4438796f6967",  
    "s3-log-project-integ-test-temp1739249999716bbd438dd-8bb8-47bd-  
ba6b-0133ac65b3d3:e2fa4eab-73af-42e3-8903-92fddaf9f378",  
    "s3-log-project-integ-test-  
temp17392474779450fbdacc2-2d6e-4190-9ad5-28f891bb7415:cd71e456-2a4c-4db4-ada5-  
da892b0bba05",  
    "s3-log-project-integ-test-temp17392246284164301421c-5030-4fa1-b4d3-  
ca15e44771c5:9e26ab3f-65e4-4896-a19c-56b1a95e630a",  
    "s3-log-project-integ-test-temp173921367319497056d8d-6d8e-4f5a-a37c-  
a62f5686731f:22d91b06-df1e-4e9c-a664-c0abb8d5920b",  
    "s3-log-project-integ-test-temp1739213439503f6283f19-390c-4dc8-95a9-  
c8480113384a:82cc413e-fc46-47ab-898f-ae23c83a613f",  
    "s3-log-project-integ-test-temp1739054385570b1f1ddc2-0a23-4062-  
bd0c-24e9e4a99b99:c02562f3-2396-42ec-98da-38e3fe5da13a",  
    "s3-log-project-integ-test-temp173905400540237dab1ac-1fde-4dfb-a8f5-  
c0114333dc89:d2f30493-f65e-4fa0-a7b6-08a5e77497b9",  
    "s3-log-project-integ-test-  
temp17390534055719c534090-7bc4-48f1-92c5-34acaec5bf1e:df5f1c8a-f017-43b7-91ba-  
ad2619e2c059",  
    "s3-log-project-integ-test-temp1739052719086a61813cc-  
ebb9-4db4-9391-7f43cc984ee4:d61917ec-8037-4647-8d52-060349272c4a",  
    "s3-log-project-integ-test-temp173898670094078b67edb-  
c42f-42ed-9db2-4b5c1a5fc66a:ce33dfbc-beeb-4466-8c99-a3734a0392c7",  
  ]  
}
```

```
"s3-log-project-integ-test-  
temp17389863425584d21b7cd-32e2-4f11-9175-72c89ecaffef:046dadf0-1f3a-4d51-a2c0-  
e88361924acf",  
  "s3-log-project-integ-test-  
temp1738985884273977ccd23-394b-46cc-90d3-7ab94cf764dc:0370dc41-9339-4b0a-91ed-51929761b244",  
  "s3-log-project-integ-test-temp1738985365972241b614f-8e41-4387-  
bd25-2b8351fbc9e0:076c392a-9630-47d8-85a9-116aa34edfff",  
  "s3-log-project-integ-test-  
temp1738985043988a51a9e2b-09d6-4d24-9c3c-1e6e21ac9fa8:6ea3949c-435b-4177-  
aa4d-614d5956244c",  
  "s3-log-project-integ-test-temp1738984123354c68b31ad-49d1-4f4b-981d-  
b66c00565ff6:6c3fff6c-815b-48b5-ada3-737400a6dee8",  
  "s3-log-project-integ-test-  
temp1738977263715d4d5bf6c-370a-48bf-8ea6-905358a6cf92:968a0f54-724a-42d1-9207-6ed854b2fae8",  
  "s3-log-project-integ-test-  
temp173897358796816ce8d7d-2a5e-41ef-855b-4a94a8d2795d:80f9a7ce-930a-402e-934e-  
d8b511d68b04",  
  "s3-log-project-integ-test-temp17389730633301af5e452-0966-467c-  
b684-4e36d47f568c:cabbe989-2e8a-473c-af25-32edc8c28646",  
  "s3-log-project-integ-test-temp1738901503813173fd468-  
b723-4d7b-9f9f-82e88d17f264:f7126a4a-b0d5-452f-814c-fea73718f805",  
  "s3-log-project-integ-test-temp1738890502472c13616fb-  
bd0f-4253-86cc-28b74c97a0ba:c6f197e5-3a53-45b6-863e-0e6353375437",  
  "s3-log-project-integ-test-  
temp17388903044683610daf3-8da7-43c6-8580-9978432432ce:d20aa317-8838-4966-  
bbfc-85b908213df1",  
  "s3-log-project-integ-test-temp173888857196780b5ab8b-e54b-44fd-a222-  
c5a374fffe96:ab4b9970-ffae-47a0-b3a8-7b6790008cad",  
  "s3-log-project-integ-test-temp1738888336931c11d378d-e74d-49a4-  
a723-3b92e6f7daac:4922f0e8-9b7d-4119-9c9f-115cd85e703e",  
  "s3-log-project-integ-test-temp17388881717651612a397-c23f-4d88-  
ba87-2773cd3fc0c9:be91c3fc-418e-4feb-8a3a-ba58ff8f4e8a",  
  "s3-log-project-integ-test-  
temp17388879727174c3c62ed-6195-4afb-8a03-59674d0e1187:a48826a8-3c0d-43c5-  
a1b5-1c98a0f978e9",  
  "s3-log-project-integ-test-temp1738885948597cef305e4-b8b4-46b0-a65b-  
e2d0a7b83294:c050e77d-e3f8-4829-9a60-46149628fe96",  
  "s3-log-project-integ-test-temp173888561463001a7d2a8-  
e4e4-4434-94db-09d3da9a9e17:8c3ac3f5-7111-4297-aec9-2470d3ead873",  
  "s3-log-project-integ-test-  
temp1738869855076eb19cafd-04fe-41bd-8aa0-40826d0c0d27:d25be134-05cb-404a-85da-  
ac5f85d2d72c",  
  "s3-project-integ-test-temp1738868157467148eacfc-d39b-49fc-a137-  
e55381cd2978:4909557b-c221-4814-b4b6-7d9e93d37c35",
```

```

        "s3-project-integ-test-temp1738820926895abec0af2-
e33d-473c-9cf4-2122dd9d6876:8f5cf218-71d6-40a4-a4be-6cacebd7765f",
        "s3-project-integ-test-temp173881998877574f969a6-1c2e-4441-b463-
ab175b45ce32:04396851-c901-4986-9117-585528e3877f",
        "s3-project-integ-test-temp17388189812309abd2604-29ba-4cf6-
b6bf-073207b7db9c:540075c7-f5ec-41e8-9341-2233c09247eb",
        "s3-project-integ-test-temp1738818843474d3ea9ac1-b609-461b-
bbdb-2da245c9bc96:865d4c3c-fbfe-4ece-9c92-d0c928341404",
        "s3-project-integ-test-temp1738818542236006e9169-e6d9-4344-9b59-
f557e7aec619:1f9ffa87-da15-4290-83e2-eebdd877497b",
        "s3-project-integ-test-
temp173881809557486ad11fd-7931-48d7-81d5-499cea52a6bc:c4c2efc4-685f-4e13-8b0f-1ef85ec300b1",
        "s3-project-integ-test-temp173881794103322941020-3f0b-49c3-b836-
fcd818ec9484:0344cfba-de48-456d-b2a8-6566bd4a5d6e",
        "s3-project-integ-test-temp1738817680747b93d0d0b-ea16-497f-9559-
af25ee6dcfdf:654a3a55-d92a-4dc6-8da8-56fd4d40d7e1",
        "s3-project-integ-test-temp17388174027191255c3da-086c-4270-b047-
acac0b7bee0d:b7e82740-2c69-42fc-ab5a-dbf15bc016a1",
        "s3-project-integ-test-temp1738817099799016e7fa3-b9b5-46a2-
bcd5-0888c646743f:8705a6a4-79ff-427a-a1c3-85c4e8fe462e",
        "s3-project-integ-test-temp1738816479281bb0c3606-5ebf-4623-
bed5-12b60e9d3512:f23fc74b-a981-4835-8e28-375fcd4c99e4",
        "s3-project-integ-test-
temp1738816263585c939a133-4d37-482c-9238-1dbff34b7674:ca28e234-0045-4ae6-8732-938b17597f50",
        "s3-project-integ-test-
temp173881580873072d18733-8fe4-43b1-83f7-95f25bb27ccf:c6f0f55b-5736-47c7-
a3aa-1b8461a6d5ed"
    ]
}

```

Tutoriel : Connexion à un sandbox via SSH

Ce didacticiel explique comment se connecter à un CodeBuild sandbox à l'aide d'un client SSH.

Pour effectuer ce didacticiel, vous devez d'abord :

- Assurez-vous d'avoir un AWS CodeBuild projet existant.
- Configurez les autorisations IAM appropriées configurées pour votre rôle dans le CodeBuild projet.
- Installez et configurez AWS CLI sur votre machine locale.

Étape 1 : démarrer un bac à sable

Pour démarrer un CodeBuild bac à sable dans la console

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération. Choisissez le projet de génération, puis choisissez Debug build.

The screenshot shows the AWS CodeBuild console interface for a project named 'sandbox-project'. At the top, there's a breadcrumb trail: 'Developer Tools > CodeBuild > Build projects > sandbox-project'. Below this, the project name 'sandbox-project' is displayed alongside several action buttons: 'Actions', 'Create trigger', 'Edit', 'Clone', 'Debug build', 'Start build with overrides', and a prominent orange 'Start build' button. A 'Configuration' section follows, containing a table with the following details:

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::012345678910:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

Below the configuration table, there are tabs for 'Build history', 'Batch history', 'Project details' (which is selected), 'Build triggers', 'Metrics', and 'Debug sessions'. The 'Project details' tab shows 'Project configuration' with an 'Edit' button. The configuration table lists:

Name	Description
sandbox-project	-
Project ARN	Build badge
arn:aws:codebuild:us-east-1:012345678910:project/sandbox-project	Disabled

3. Dans l'onglet Client SSH, choisissez Start sandbox.

The screenshot shows the 'Debug build' page in the AWS CodeBuild console. The breadcrumb trail is 'Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build'. The page title is 'Debug build'. There are three tabs: 'Run Command', 'SSH Client' (which is selected), and 'Session Manager'. Below the tabs, there's a light blue information box with an 'i' icon and the text 'Connect to your SSH client with sandbox'. It includes two bullet points:

- Launches a sandbox environment with SSH connectivity.
- Connect directly using SSH clients or your preferred IDE.

A 'Learn more' link with an external icon is located to the right of the information box. At the bottom right of the page, there is an orange 'Start sandbox' button.

4. Le processus d'initialisation du sandbox peut prendre un certain temps. Vous pouvez vous connecter au sandbox lorsque son statut passe à RUN_SANDBOX.

Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build

Debug build

Run CommandSSH ClientSession Manager

Sandbox is running

Your sandbox `sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256` is ready and available for use.

Stop sandbox

TerminalVisual Studio CodeIntelliJ IDEA

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Make sure your CLI user has the `codebuild:StartSandboxConnection` permission. For more information, see [AWS CLI authentication](#) documentation.

Connect to your sandbox environment with following command:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:012345678910:sandbox/sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256
```

Sandbox phasesSandbox logsSandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
QUEUED	Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
PROVISIONING	Succeeded		4 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	Succeeded		6 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
RUN_SANDBOX	-	-	-	Apr 1, 2025 4:33 PM (UTC-7:00)	-

Étape 2 : Modifier la configuration SSH locale

Si vous vous connectez au sandbox pour la première fois, vous devez effectuer un processus de configuration unique en suivant les étapes suivantes :

Pour modifier la configuration SSH locale dans la console

1. Localisez les commandes de configuration de votre système d'exploitation.

2. Ouvrez votre terminal local, puis copiez et exécutez les commandes fournies pour télécharger et exécuter le script permettant de configurer votre configuration SSH locale. Par exemple, si votre système d'exploitation est macOS, utilisez la commande suivante :

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Déboguer les builds avec Sandbox CodeBuild

Version de l'API 2016-10-06 703

3. Le script de configuration ajoutera les configurations requises pour la connexion à vos sandbox. Vous serez invité à accepter ces modifications.
4. Une fois la configuration réussie, une nouvelle entrée de configuration SSH pour le CodeBuild sandbox sera créée.

```
Host codebuild-sandbox-ssh*
  StrictHostKeyChecking no
  LogLevel INFO
  ForwardAgent yes
  ControlMaster auto
  ControlPersist 10m
  ProxyCommand sh -c "/Users/[redacted]/.aws/codebuild-dev-env/codebuild-sandbox-connect.sh %n"
```

Étape 3 : Connectez-vous au bac à sable

Pour modifier la configuration SSH locale dans la console

1. Configurez AWS CLI l'authentification et assurez-vous que votre AWS CLI utilisateur dispose de l'`codebuild:StartSandboxConnection` autorisation. Pour plus d'informations, consultez la section [Authentification à l'aide des informations d'identification utilisateur IAM AWS CLI dans le guide de l'utilisateur de l'interface de ligne de commande pour la version 1](#).
2. Connectez-vous à votre sandbox à l'aide de la commande suivante :

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>
```

Note

Pour résoudre les problèmes de connexion, utilisez l'-v indicateur pour activer la sortie détaillée. Par exemple, `ssh -v codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>`. Pour obtenir des conseils de dépannage supplémentaires, consultez [Résolution des problèmes AWS CodeBuild de connexion SSH au sandbox](#).

Étape 4 : Passez en revue vos résultats

Une fois connecté, vous pouvez corriger les échecs de compilation, tester les commandes de compilation, expérimenter des modifications de configuration et vérifier les variables d'environnement et les dépendances avec votre sandbox.

Résolution des problèmes AWS CodeBuild de connexion SSH au sandbox

Utilisez les informations de cette rubrique pour identifier, diagnostiquer et résoudre les problèmes de connexion SSH dans le CodeBuild sandbox.

Rubriques

- [StartSandboxConnectionInvalidInputException erreur lors du SSH dans un environnement CodeBuild sandbox](#)
- [Erreur : « Impossible de localiser les informations d'identification » lors du SSH dans un environnement CodeBuild sandbox](#)
- [StartSandboxConnectionAccessDeniedException erreur lors du SSH dans un environnement CodeBuild sandbox](#)
- [Erreur : « ssh : Impossible de résoudre le nom d'hôte » lors du SSH dans un environnement sandbox CodeBuild](#)

StartSandboxConnectionInvalidInputException erreur lors du SSH dans un environnement CodeBuild sandbox

Problème : Lorsque vous tentez de vous connecter à un environnement CodeBuild sandbox à l'aide de la commande `ssh codebuild-sandbox-ssh=<sandbox-arn>`, vous pouvez rencontrer une `InvalidInputException` erreur telle que :

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for {sandbox-arn}
User: arn:aws:sts::<account-ID>:assumed-role/<service-role-name>/AWSCodeBuild-<UUID>
is not authorized to perform: ssm:StartSession on resource.
```

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for
sandbox <sandbox-arn>: codebuild:<UUID> is not connected.
```

Cause possible :

- Agent Amazon EC2 Systems Manager manquant : l'agent SSM n'est pas correctement installé ou configuré sur l'image de compilation.
- Autorisations insuffisantes : le rôle de service de CodeBuild projet ne dispose pas des autorisations SSM requises.

Solution recommandée : Si vous utilisez une image personnalisée pour votre build, procédez comme suit.

1. Installez SSM Agent. Pour plus d'informations, consultez la section [Installation et désinstallation manuelles de l'agent SSM sur les instances Amazon EC2 pour Linux](#) dans le. La version de l'agent SSM doit être 3.0.1295.0 ou ultérieure.
2. Copiez le fichier <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/7.0/amazon-ssm-agent.json> dans le /etc/amazon/ssm/ répertoire de votre image. Cela active le mode conteneur dans l'agent SSM.
3. Assurez-vous que le rôle de service de votre CodeBuild projet dispose des autorisations suivantes, puis redémarrez l'environnement sandbox :

```
{
  "Effect": "Allow",
  "Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
    "ssmmessages:OpenDataChannel"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:build/*",
    "arn:aws:ssm:region::document/AWS-StartSSHSession"
  ]
}
```

Erreur : « Impossible de localiser les informations d'identification » lors du SSH dans un environnement CodeBuild sandbox

Problème : Lorsque vous tentez de vous connecter à un environnement CodeBuild sandbox à l'aide de la commande `ssh codebuild-sandbox-ssh=<sandbox-arn>`, vous pouvez rencontrer l'erreur d'identification suivante :

Unable to locate credentials. You can configure credentials by running "aws configure".

Cause possible : les AWS informations d'identification n'ont pas été correctement configurées dans votre environnement local.

Solution recommandée : Configurez vos AWS CLI informations d'identification en suivant la documentation officielle : [Configuration des paramètres du Guide de AWS CLI l'utilisateur](#) de l'interface de ligne de AWS commande pour la version 2.

StartSandboxConnectionAccessDeniedException erreur lors du SSH dans un environnement CodeBuild sandbox

Problème : Lorsque vous tentez de vous connecter à un environnement CodeBuild sandbox à l'aide de la commande `ssh codebuild-sandbox-ssh=<sandbox-arn>`, vous pouvez rencontrer l'erreur d'autorisation suivante :

```
An error occurred (AccessDeniedException) when calling the StartSandboxConnection operation:
User: arn:aws:sts::account-id:assumed-role/role-name
is not authorized to perform: codebuild:StartSandboxConnection on resource:
sandbox-arn
because no identity-based policy allows the codebuild:StartSandboxConnection action
```

Cause possible : Vos AWS informations d'identification ne disposent pas CodeBuild des autorisations nécessaires pour effectuer cette opération.

Solution recommandée : assurez-vous que l'utilisateur ou le rôle IAM associé à vos AWS CLI informations d'identification dispose des autorisations suivantes :

```
{
  "Effect": "Allow",
  "Action": [
    "codebuild:StartSandboxConnection"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:sandbox/*"
  ]
}
```

Erreur : « ssh : Impossible de résoudre le nom d'hôte » lors du SSH dans un environnement sandbox CodeBuild

Problème : Lorsque vous tentez de vous connecter à un environnement CodeBuild sandbox à l'aide de la commande `ssh codebuild-sandbox-ssh=<sandbox-arn>`, vous rencontrez l'erreur de résolution de nom d'hôte suivante :

```
ssh: Could not resolve hostname
```

Cause possible : Cette erreur se produit généralement lorsque le script de connexion au CodeBuild sandbox requis n'a pas été correctement exécuté dans votre environnement local.

Solution recommandée:

1. Téléchargez le script de connexion au CodeBuild sandbox.
2. Exécutez le script dans votre terminal pour établir la configuration SSH nécessaire.
3. Réessayez de vous connecter SSH à l'environnement sandbox.

Déboguer les builds avec le gestionnaire de session

Dans AWS CodeBuild, vous pouvez suspendre une compilation en cours, puis utiliser le gestionnaire de AWS Systems Manager session pour vous connecter au conteneur de génération et afficher l'état du conteneur.

Note

Cette fonctionnalité n'est pas disponible dans les environnements Windows.

Rubriques

- [Conditions préalables](#)
- [Suspendre le build](#)
- [Commencez la construction](#)
- [Connectez-vous au conteneur de construction](#)
- [Reprendre le build](#)

Conditions préalables

Pour autoriser l'utilisation du gestionnaire de session avec la session de génération, vous devez activer la connexion à la session pour la génération. Il existe deux prérequis :

- CodeBuild L'agent SSM est déjà installé et l'agent SSM est activé sur les images sélectionnées standard de Linux. ContainerMode

Si vous utilisez une image personnalisée pour votre build, procédez comme suit :

1. Installez SSM Agent. Pour plus d'informations, consultez la section [Installation manuelle de l'agent SSM sur les instances EC2 pour Linux](#) dans le guide de l' AWS Systems Manager utilisateur. La version de l'agent SSM doit être 3.0.1295.0 ou ultérieure.
2. Copiez le fichier <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent.json> dans le /etc/amazon/ssm/ répertoire de votre image. Cela active le mode conteneur dans l'agent SSM.

Note

Les images personnalisées nécessiteraient la mise à jour de l'agent SSM le plus récent pour que cette fonctionnalité fonctionne comme prévu.

- Le rôle CodeBuild de service doit respecter la politique SSM suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
    }
  ]
}
```



```
}
```

Vous pouvez demander à la CodeBuild console d'associer automatiquement cette politique à votre rôle de service lorsque vous démarrez la génération. Vous pouvez également associer cette politique à votre rôle de service manuellement.

- Si l'audit et la journalisation de l'activité des sessions sont activés dans les préférences de Systems Manager, le rôle de CodeBuild service doit également disposer d'autorisations supplémentaires. Les autorisations sont différentes en fonction de l'endroit où les journaux sont stockés.

CloudWatch Journaux

Si vous utilisez CloudWatch Logs pour stocker vos journaux, ajoutez l'autorisation suivante au rôle CodeBuild de service :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:MyLogGroup:*"
    }
  ]
}
```

Amazon S3

Si vous utilisez Amazon S3 pour stocker vos journaux, ajoutez l'autorisation suivante au rôle CodeBuild de service :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetEncryptionConfiguration",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Pour plus d'informations, consultez la section [Audit et journalisation de l'activité des sessions](#) dans le guide de AWS Systems Manager l'utilisateur.

Suspendre le build

Pour suspendre la génération, insérez la `codebuild-breakpoint` commande dans l'une des phases de construction de votre fichier `buildspec`. La construction sera suspendue à ce stade, ce qui vous permettra de vous connecter au conteneur de construction et de visualiser le conteneur dans son état actuel.

Par exemple, ajoutez ce qui suit aux phases de construction dans votre fichier `buildspec`.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Ce code crée le `/tmp/hello-world` fichier, puis suspend la compilation à ce stade.

Commencez la construction

Pour autoriser l'utilisation du gestionnaire de session avec la session de génération, vous devez activer les connexions de session pour la génération. Pour ce faire, lors du démarrage de la compilation, procédez comme suit :

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération. Choisissez le projet de construction, puis choisissez Démarrer la construction avec des remplacements.
3. Choisissez Advanced build overrides (Remplacements avancés de génération).
4. Dans la section Environnement, choisissez l'option Activer la connexion à la session. Si cette option n'est pas sélectionnée, toutes les codebuild-resume commandes codebuild-breakpoint et sont ignorées.
5. Apportez les autres modifications souhaitées, puis choisissez Démarrer la construction.
6. Surveillez l'état de construction dans la console. Lorsque la session est disponible, le lien du gestionnaire de AWS session apparaît dans la section État de la création.

Connectez-vous au conteneur de construction

Vous pouvez vous connecter au conteneur de construction de deux manières :

CodeBuild console

Dans un navigateur Web, ouvrez le lien du gestionnaire de AWS session pour vous connecter au conteneur de construction. Une session de terminal s'ouvre pour vous permettre de parcourir et de contrôler le conteneur de construction.

AWS CLI

Note

Le plug-in Session Manager doit être installé sur votre machine locale pour cette procédure. Pour plus d'informations, voir [Installer le plug-in du gestionnaire de session pour la AWS CLI](#) dans le guide de AWS Systems Manager l'utilisateur.

1. Appelez l'batch-get-builds API avec l'ID de build pour obtenir des informations sur le build, y compris l'identifiant cible de session. Le nom de propriété de l'identifiant cible de session varie en fonction du type de sortie de la aws commande. C'est pourquoi --output json il est ajouté à la commande.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Copiez la valeur sessionTarget de la propriété. Le nom de la sessionTarget propriété peut varier en fonction du type de sortie de la aws commande. C'est pourquoi --output json il est ajouté à la commande à l'étape précédente.
3. Utilisez la commande suivante pour vous connecter au conteneur de construction.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Pour cet exemple, vérifiez que le /tmp/hello-world fichier existe et contient le texteHello World.

Reprendre le build

Une fois que vous avez terminé d'examiner le conteneur de construction, émettez la codebuild-resume commande depuis le shell du conteneur.

```
$ codebuild-resume
```

Supprimer les builds AWS CodeBuild

Vous pouvez utiliser le AWS CLI ou le AWS SDKs pour supprimer des versions intégrales AWS CodeBuild.

Rubriques

- [Suppression de générations \(AWS CLI\)](#)
- [Suppression de générations \(AWS SDKs\)](#)

Suppression de générations (AWS CLI)

Exécutez la commande batch-delete-builds :

```
aws codebuild batch-delete-builds --ids ids
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- *ids*: chaîne obligatoire. Les IDs versions à supprimer. Pour spécifier plusieurs générations, séparez les ID par un espace. Pour obtenir la liste des versions IDs, consultez les rubriques suivantes :
 - [Afficher la liste des build IDs \(AWS CLI\)](#)
 - [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#)

En cas de réussite, un tableau `buildsDeleted` s'affiche dans la sortie avec l'Amazon Resource Name (ARN) de chaque génération supprimée. Des informations sur les générations qui n'ont pas pu être supprimées apparaissent dans la sortie dans un tableau `buildsNotDeleted`.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Des informations similaires à ce qui suit s'affichent dans la sortie :

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
  ]
}
```

Suppression de générations (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Réessayer les builds manuellement AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs réessayer manuellement une génération unique ou une intégration par lots. AWS CLI AWS CodeBuild

Rubriques

- [Réessayer une compilation manuellement \(console\)](#)
- [Réessayer une compilation manuellement \(\)AWS CLI](#)
- [Réessayer une compilation manuellement \(\)AWS SDKs](#)

Réessayer une compilation manuellement (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si la **build-ID** page **build-project-name**: s'affiche, choisissez Réessayer de construire.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des versions, cochez la case correspondant à la génération, puis choisissez Réessayer de construire.
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de génération. Dans la liste des versions, cochez la case correspondant à la génération, puis choisissez Réessayer de construire.

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Réessayer une compilation manuellement ()AWS CLI

- Exécutez la commande `retry-build` :

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- **<build-id>**: chaîne obligatoire. L'ID de la version ou de la génération par lots à réessayer. Pour obtenir la liste des versions IDs, consultez les rubriques suivantes :
 - [Afficher la liste des build IDs \(AWS CLI\)](#)
 - [Afficher la liste des build IDs \(AWS CLI\) par lots](#)
 - [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#)
 - [Afficher la liste des compilations par lots IDs pour un projet de construction \(AWS CLI\)](#)
- **--idempotency-token**: Facultatif. Si vous exécutez la `retry-build` commande avec l'option, un identifiant unique distinguant majuscules et minuscules, ou jeton, est inclus dans la `retry-build` demande. Le jeton est valide pendant 5 minutes après la demande . Si vous répétez la `retry-build` demande avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance des paramètres.

Réessayer une compilation manuellement (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Retry s'intègre automatiquement AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs réessayer automatiquement vos builds. AWS CLI AWS CodeBuild Lorsque le réessai automatique est activé, CodeBuild il appellera automatiquement `RetryBuild` en utilisant le rôle de service du projet après un échec de construction jusqu'à une limite spécifiée. Par exemple, si la limite de tentatives automatiques est définie sur deux, l'API CodeBuild appellera l'`RetryBuildAPI` pour réessayer automatiquement votre build jusqu'à deux fois supplémentaires.

Note

CodeBuild ne prend pas en charge la nouvelle tentative automatique pour. CodePipeline

Rubriques

- [Réessayer une compilation automatiquement \(console\)](#)
- [Réessayer une compilation automatiquement \(\)AWS CLI](#)
- [Réessayer automatiquement un build \(\)AWS SDKs](#)

Réessayer une compilation automatiquement (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Sélectionnez Create a project (Créer un projet). Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#).
 - Dans Environment (Environnement) :
 - Pour Limite de tentatives automatiques, entrez le nombre maximum de tentatives automatiques souhaitées après un échec de compilation.
3. Dans Environnement, sélectionnez Configuration supplémentaire.
4. Continuez avec les valeurs par défaut, puis choisissez Create build project.

Réessayer une compilation automatiquement ()AWS CLI

- Exécutez la commande create-project :

```
aws codebuild create-project \  
  --name "<project-name>" \  
  --auto-retry-limit <auto-retry-limit> \  
  --source "<source>" \  
  --artifacts {<artifacts>} \  
  --environment "{\"type\": \"<environment-type>\", \"image\": \"<image-type>\",  
  \"computeType\": \"<compute-type>\"}" \  
  --service-role "<service-role>"
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- **<auto-retry-limit>**: Définissez la limite de tentatives automatiques au nombre maximum de tentatives automatiques souhaité après l'échec d'une compilation.

- `<project-name>`, `<source>`, `<artifacts>`, `environment-type`, `image-type` `compute-type`, et `service-role` : définissez les paramètres de configuration de projet souhaités.

Réessayer automatiquement un build ()AWS SDKs

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Arrêter les intégrations AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console AWS CLI, ou AWS SDKs pour arrêter une intégration AWS CodeBuild.

Rubriques

- [Arrêt d'une génération \(console\)](#)
- [Arrêt d'une génération \(AWS CLI\)](#)
- [Arrêt d'une génération \(AWS SDKs\)](#)

Arrêt d'une génération (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si la **build-ID** page **build-project-name** s'affiche, choisissez Arrêter la construction.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).

 Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Si vous AWS CodeBuild ne parvenez pas à arrêter une génération (par exemple, si le processus de génération est déjà terminé), le bouton Arrêter est désactivé ou risque de ne pas apparaître.

Arrêt d'une génération (AWS CLI)

- Exécutez la commande stop-build :

```
aws codebuild stop-build --id id
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- id*: chaîne obligatoire. ID de la génération à arrêter. Pour obtenir la liste des versions IDs, consultez les rubriques suivantes :
 - [Afficher la liste des build IDs \(AWS CLI\)](#)
 - [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#)

En cas d'arrêt AWS CodeBuild réussi de la construction, la buildStatus valeur de l'buildobjet dans la sortie estSTOPPED.

Si vous CodeBuild ne parvenez pas à arrêter la construction (par exemple, si la génération est déjà terminée), la buildStatus valeur de l'buildobjet dans la sortie est le statut final de la construction (par exemple,SUCCEEDED).

Arrêt d'une génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Arrêter les compilations par lots AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs arrêter l'intégration d'un lot AWS CodeBuild. AWS CLI

Note

Si vous utilisez le calcul Lambda dans votre génération par lots, la génération Lambda en cours ne peut pas être arrêtée.

Rubriques

- [Arrêter une génération par lots \(console\)](#)
- [Arrêter une construction par lots \(AWS CLI\)](#)
- [Arrêter une construction par lots \(AWS SDKs\)](#)

Arrêter une génération par lots (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si la **build-ID** page **build-project-name** s'affiche, choisissez Arrêter la construction.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme

d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Arrêter une construction par lots (AWS CLI)

- Exécutez la commande [stop-build-batch](#) :

```
aws codebuild stop-build-batch --id <batch-build-id>
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- <batch-build-id>**: chaîne obligatoire. Identifiant du lot à arrêter. Pour obtenir la liste des identifiants de génération par lots, consultez les rubriques suivantes :
 - [Afficher la liste des build IDs \(AWS CLI\) par lots](#)
 - [Afficher la liste des compilations par lots IDs pour un projet de construction \(AWS CLI\)](#)

Arrêter une construction par lots (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Le déclencheur AWS CodeBuild se construit automatiquement

Vous pouvez créer un déclencheur sur un projet pour planifier une génération une fois toutes les heures, tous les jours ou toutes les semaines. Vous pouvez également modifier un déclencheur pour utiliser une règle personnalisée avec une expression Amazon CloudWatch Cron. Par exemple, à l'aide d'une expression cron, vous pouvez planifier une génération à une heure précise chaque jour de la semaine. Pour plus d'informations sur la création et la modification de déclencheurs, reportez-vous [Création de AWS CodeBuild déclencheurs](#) aux sections et [Modifier les AWS CodeBuild déclencheurs](#).

Rubriques

- [Création de AWS CodeBuild déclencheurs](#)
- [Modifier les AWS CodeBuild déclencheurs](#)

Création de AWS CodeBuild déclencheurs

Vous pouvez créer un déclencheur sur un projet pour planifier une génération une fois toutes les heures, tous les jours ou toutes les semaines. Vous pouvez également créer un déclencheur à l'aide d'une règle personnalisée avec une expression Amazon CloudWatch cron. Par exemple, à l'aide d'une expression cron, vous pouvez planifier une génération à une heure précise chaque jour de la semaine.

Note

Il n'est pas possible de démarrer une génération par lots à partir d'un déclencheur de génération, d'un EventBridge événement Amazon ou d'une AWS Step Functions tâche.

Rubriques

- [Création de AWS CodeBuild déclencheurs \(console\)](#)
- [Création de AWS CodeBuild déclencheurs par programmation](#)

Création de AWS CodeBuild déclencheurs (console)

Utilisez la procédure suivante pour créer des déclencheurs à l'aide du AWS Management Console.

Pour créer un déclencheur

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Sélectionnez le lien pour le projet de génération pour lequel vous voulez ajouter un déclencheur, puis choisissez l'onglet Build triggers (Déclencheurs de génération).

Note

Par défaut, les 100 projets de construction les plus récents sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

4. Choisissez Create trigger.
5. Saisissez un nom dans Trigger name (Nom du déclencheur).
6. Dans la liste déroulante Fréquence, choisissez la fréquence de votre déclencheur. Si vous voulez créer une fréquence à l'aide d'une expression cron, choisissez Custom (Personnalisé).
7. Spécifiez les paramètres de la fréquence de votre déclencheur. Vous pouvez saisir les tout premiers caractères de vos sélections dans la zone de texte pour filtrer les éléments du menu déroulant.

 Note

Les heures et minutes de début sont basées sur zéro. La minute de début est un nombre compris entre zéro et 59. L'heure de début est un nombre compris entre zéro et 23. Par exemple, un déclencheur quotidien qui commence tous les jours à 12 h 15 a une heure de début de 12 heures et une minute de début de 15. Un déclencheur quotidien qui commence tous les jours à minuit a une heure de début de zéro et une minute de début de zéro. Un déclencheur quotidien qui commence tous les jours à 23 h 59 a une heure de début de 23 et une minute de départ de 59.

Fréquence	Paramètres requis	Détails
Par heure	Minute de début	Utilisez le menu déroulant Start minute (Minute de début).
Chaque jour	Minute de début Heure de début	Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début).
Hebdomadaire	Minute de début Heure de début Jour de début	Utilisez le menu déroulant Start minute (Minute de début).

Fréquence	Paramètres requis	Détails
		Utilisez le menu déroulant Start hour (Heure de début).
		Utilisez le menu déroulant Start day (Jour de début).
Personnalisé	Expression cron	Saisissez une expression cron dans Cron expression (Expression Cron). Une expression cron est composée de six champs obligatoires qui sont séparés par des espaces. Les champs spécifient une valeur de début pour la minute, l'heure, le jour du mois, le mois, le jour de la semaine et l'année. Vous pouvez utiliser des caractères génériques pour spécifier une plage, des valeurs supplémentaires, etc. Par exemple, l'expression cron <code>0 9 ? * MON-FRI *</code> planifie une génération tous les jours de la semaine à 9 h 00. Pour plus d'informations, consultez la section Cron Expressions dans le guide de l'utilisateur Amazon CloudWatch Events.

8. Sélectionnez **Enable this trigger** (Activer ce déclencheur).
9. (Facultatif) Développez **Advanced section** (Section avancée). Dans **Source version** (Version de la source), saisissez une version de votre code source.

- Pour Amazon S3, entrez l'ID de version qui correspond à la version de l'artefact d'entrée que vous souhaitez créer. Si le champ Version de la source reste vide, la dernière version est utilisée.
 - Pour AWS CodeCommit, saisissez un ID de validation. Si Version de la source est vide, l'ID de validation HEAD de la branche par défaut est utilisé.
 - Pour GitHub ou GitHub Enterprise, saisissez un identifiant de validation, un identifiant de pull request, un nom de branche ou un nom de balise correspondant à la version du code source que vous souhaitez créer. Si vous spécifiez un ID de demande d'extraction, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si vous spécifiez un nom de branche, l'ID de validation HEAD de la branche est utilisé. Si Version source est vide, l'ID de validation HEAD de la branche par défaut est utilisé.
 - Pour Bitbucket, entrez un ID de validation, un nom de branche ou un nom de balise qui correspond à la version du code source à générer. Si vous spécifiez un nom de branche, l'ID de validation HEAD de la branche est utilisé. Si Version source est vide, l'ID de validation HEAD de la branche par défaut est utilisé.
10. (Facultatif) Spécifiez un délai compris entre 5 minutes et 2 160 minutes (36 heures). Cette valeur indique la durée des AWS CodeBuild tentatives de compilation avant qu'elle ne s'arrête. Si les Hours (heures) et Minutes sont laissées vides, la valeur de délai d'expiration par défaut spécifiée dans le projet est utilisée.
11. Choisissez Create trigger.

Création de AWS CodeBuild déclencheurs par programmation

CodeBuild utilise les EventBridge règles Amazon pour créer des déclencheurs. Vous pouvez utiliser l' EventBridge API pour créer par programmation des déclencheurs de build pour vos CodeBuild projets. Consultez le [manuel Amazon EventBridge API Reference](#) pour plus d'informations.

Modifier les AWS CodeBuild déclencheurs

Vous pouvez modifier un déclencheur sur un projet pour planifier une génération une fois toutes les heures, tous les jours ou toutes les semaines. Vous pouvez également modifier un déclencheur pour utiliser une règle personnalisée avec une expression Amazon CloudWatch Cron. Par exemple, à l'aide d'une expression cron, vous pouvez planifier une génération à une heure précise chaque jour de la semaine. Pour plus d'informations sur la création d'un déclencheur, consultez [Création de AWS CodeBuild déclencheurs](#).

Rubriques

- [Modifier les AWS CodeBuild déclencheurs \(console\)](#)
- [Modifier les AWS CodeBuild déclencheurs par programmation](#)

Modifier les AWS CodeBuild déclencheurs (console)

Utilisez la procédure suivante pour modifier les déclencheurs à l'aide du AWS Management Console.

Pour modifier un déclencheur

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Sélectionnez le lien du projet de génération à modifier, puis choisissez l'onglet Build triggers (Déclencheurs de génération).

Note

Par défaut, les 100 projets de construction les plus récents sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

4. Sélectionnez la case d'option en regard du déclencheur à modifier, puis choisissez Edit (Modifier).
5. Dans la liste déroulante Fréquence, choisissez la fréquence de votre déclencheur. Si vous voulez créer une fréquence à l'aide d'une expression cron, choisissez Custom (Personnalisé).
6. Spécifiez les paramètres de la fréquence de votre déclencheur. Vous pouvez saisir les tout premiers caractères de vos sélections dans la zone de texte pour filtrer les éléments du menu déroulant.

Note

Les heures et minutes de début sont basées sur zéro. La minute de début est un nombre compris entre zéro et 59. L'heure de début est un nombre compris entre zéro et 23. Par exemple, un déclencheur quotidien qui commence tous les jours à 12 h 15 a une heure de début de 12 heures et une minute de début de 15. Un déclencheur quotidien qui

commence tous les jours à minuit a une heure de début de zéro et une minute de début de zéro. Un déclencheur quotidien qui commence tous les jours à 23 h 59 a une heure de début de 23 heures et une minute de départ de 59.

Fréquence	Paramètres requis	Détails
Par heure	Minute de début	Utilisez le menu déroulant Start minute (Minute de début).
Chaque jour	Minute de début Heure de début	Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début).
Hebdomadaire	Minute de début Heure de début Jour de début	Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début). Utilisez le menu déroulant Start day (Jour de début).

Fréquence	Paramètres requis	Détails
Personnalisé	Expression cron	Saisissez une expression cron dans Cron expression (Expression Cron). Une expression cron est composée de six champs obligatoires qui sont séparés par des espaces. Les champs spécifient une valeur de début pour la minute, l'heure, le jour du mois, le mois, le jour de la semaine et l'année. Vous pouvez utiliser des caractères génériques pour spécifier une plage, des valeurs supplémentaires, etc. Par exemple, l'expression cron 0 9 ? * MON-FRI * planifie une génération tous les jours de la semaine à 9 h 00. Pour plus d'informations, consultez la section Cron Expressions dans le guide de l'utilisateur Amazon CloudWatch Events.

7. Sélectionnez **Enable this trigger** (Activer ce déclencheur).

 Note

Vous pouvez utiliser la CloudWatch console Amazon <https://console.aws.amazon.com/cloudwatch/> pour modifier la version de la source, le délai d'expiration et d'autres options qui ne sont pas disponibles dans AWS CodeBuild.

Modifier les AWS CodeBuild déclencheurs par programmation

CodeBuild utilise les EventBridge règles d'Amazon pour créer des déclencheurs. Vous pouvez utiliser l' EventBridge API pour modifier par programmation les déclencheurs de génération de vos CodeBuild projets. Consultez le [manuel Amazon EventBridge API Reference](#) pour plus d'informations.

Afficher les détails du build dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs pour afficher les détails des builds gérés par CodeBuild. AWS CLI

Rubriques

- [Affichage des détails d'une génération \(console\)](#)
- [Affichage des détails d'une génération \(AWS CLI\)](#)
- [Affichage des détails d'une génération \(AWS SDKs\)](#)
- [Transitions des phases de génération](#)

Affichage des détails d'une génération (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, dans la colonne Exécution de génération, choisissez le lien de la génération.
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien du nom du projet de génération. Puis, dans la liste des générations, dans la colonne Exécution de génération, choisissez le lien de la génération.

Note

Par défaut, seuls les 10 derniers projets de génération ou générations s'affichent. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme

d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Affichage des détails d'une génération (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Exécutez la commande batch-get-builds :

```
aws codebuild batch-get-builds --ids ids
```

Remplacez l'espace réservé suivant :

- *ids*: chaîne obligatoire. Une ou plusieurs versions sur IDs lesquelles afficher les détails. Pour spécifier plusieurs ID de génération, séparez-les par un espace. Vous pouvez spécifier jusqu'à 100 versions IDs. Pour obtenir la liste des versions IDs, consultez les rubriques suivantes :
 - [Afficher la liste des build IDs \(AWS CLI\)](#)
 - [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#)

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

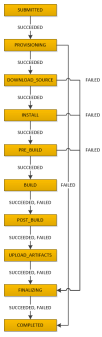
Si la commande aboutit, des données similaires à celles décrites dans [Pour afficher les informations de génération résumées](#) apparaissent dans la sortie.

Affichage des détails d'une génération (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Transitions des phases de génération

Les constructions se AWS CodeBuild déroulent par étapes :



 Important

La phase UPLOAD_ARTIFACTS est toujours tentée, même si la phase BUILD échoue.

Afficher la liste des éléments IDs intégrés AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console ou AWS SDKs afficher la liste des versions IDs pour les versions gérées par CodeBuild. AWS CLI

Rubriques

- [Afficher la liste des versions IDs \(console\)](#)
- [Afficher la liste des build IDs \(AWS CLI\)](#)
- [Afficher la liste des build IDs \(AWS CLI\) par lots](#)
- [Afficher la liste des build IDs \(AWS SDKs\)](#)

Afficher la liste des versions IDs (console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Historique de génération.

 Note

Par défaut, seules les 10 dernières générations s'affichent. Pour afficher plus de générations, sélectionnez l'icône en forme d'engrenage, puis sélectionnez une autre valeur pour Générations par page ou utilisez les flèches avant et arrière.

Afficher la liste des build IDs (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with CodeBuild, consultez le [Référence des commandes en ligne](#).

- Exécutez la commande list-builds :

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- sort-order*: chaîne facultative utilisée pour indiquer comment répertorier le build IDs. Les valeurs valides sont ASCENDING et DESCENDING.
- next-token*: chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild list-builds --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Afficher la liste des build IDs (AWS CLI) par lots

Pour plus d'informations sur l'utilisation du AWS CLI with CodeBuild, consultez le [Référence des commandes en ligne](#).

- Exécutez la commande list-build-batches :

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *sort-order*: chaîne facultative utilisée pour indiquer comment répertorier la compilation par lots IDs. Les valeurs valides sont ASCENDING et DESCENDING.
- *next-token*: chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :


```
aws codebuild list-build-batches --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Afficher la liste des build IDs (AWS SDKs)

Pour plus d'informations sur l'utilisation CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Afficher la liste des versions IDs d'un projet de construction dans AWS CodeBuild

Vous pouvez utiliser la AWS CodeBuild console AWS CLI, ou AWS SDKs pour afficher la liste des versions IDs d'un projet de génération dans CodeBuild.

Rubriques

- [Afficher la liste des builds IDs pour un projet de build \(console\)](#)
- [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#)
- [Afficher la liste des compilations par lots IDs pour un projet de construction \(AWS CLI\)](#)
- [Afficher la liste des versions IDs pour un projet de construction \(AWS SDKs\)](#)

Afficher la liste des builds IDs pour un projet de build (console)

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le projet de génération.

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Afficher la liste des versions IDs pour un projet de construction (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Exécutez la commande `list-builds-for-project`, comme suit :

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- ***project-name***: chaîne obligatoire utilisée pour indiquer le nom du projet de construction IDs pour lequel répertorier les versions. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- ***sort-order***: chaîne facultative utilisée pour indiquer comment répertorier le build IDs. Les valeurs valides sont ASCENDING et DESCENDING.
- ***next-token***: chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur renvoyé, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez cette commande comme suit :

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Afficher la liste des compilations par lots IDs pour un projet de construction (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

Exécutez la commande `list-build-batches-for-project`, comme suit :

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-
order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *project-name*: chaîne obligatoire utilisée pour indiquer le nom du projet de construction IDs pour lequel répertorier les versions. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- *sort-order*: chaîne facultative utilisée pour indiquer comment répertorier le build IDs. Les valeurs valides sont ASCENDING et DESCENDING.
- *next-token*: chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur renvoyé, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez cette commande comme suit :

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --
sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project
--sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY2OA==
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Afficher la liste des versions IDs pour un projet de construction (AWS SDKs)

Pour plus d'informations sur l'utilisation AWS CodeBuild avec le AWS SDKs, consultez le [AWS SDKs et référence des outils](#).

Rapports de test dans AWS CodeBuild

Vous pouvez créer des rapports CodeBuild contenant des détails sur les tests exécutés pendant les builds. Vous pouvez créer des tests tels que des tests unitaires, des tests de configuration et des tests fonctionnels.

Les formats de fichier de rapport de test suivants sont pris en charge :

- Concombre JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- Visual Studio TRX XML (.xml)

Note

La dernière version prise en charge de `cucumber-js` est la 7.3.2.

Créez vos scénarios de test avec n'importe quel framework de test capable de créer des fichiers de rapport dans l'un de ces formats (par exemple, le JUnit plugin Surefire, TestNG ou Cucumber).

Pour créer un rapport de test, ajoutez un nom de groupe de rapports au fichier `buildspec` d'un projet de construction avec des informations sur vos cas de test. Lorsque vous exécutez le projet de construction, les cas de test sont exécutés et un rapport de test est créé. Un nouveau rapport de test est créé dans le groupe de rapports chaque fois que les scénarios de test sont exécutés. Vous n'avez pas besoin de créer un groupe de rapports avant d'exécuter vos tests. Si vous spécifiez un nom de groupe de rapports, CodeBuild crée un groupe de rapports pour vous lorsque vous générez vos rapports. Si vous souhaitez utiliser un groupe de rapports qui existe déjà, vous spécifiez son ARN dans le fichier `buildspec`.

Vous pouvez utiliser un rapport de test pour résoudre un problème lors d'une génération. Si vous disposez de nombreux rapports de test provenant de plusieurs versions d'un projet de construction,

vous pouvez utiliser vos rapports de test pour afficher les tendances et les taux de test et d'échec afin de vous aider à optimiser les versions.

Un rapport expire 30 jours après sa création. Vous ne pouvez pas afficher un rapport de test expiré. Si vous souhaitez conserver les rapports de test pendant plus de 30 jours, vous pouvez exporter les fichiers de données brutes de vos résultats de test vers un compartiment Amazon S3. Les fichiers de test exportés n'expirent pas. Des informations sur le compartiment S3 sont spécifiées lorsque vous créez le groupe de rapports.

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Rubriques

- [Création de rapports de test](#)
- [Création de rapports sur la couverture du code](#)
- [Découvrez automatiquement les rapports dans CodeBuild](#)
- [Groupes de rapports](#)
- [Cadres de test](#)
- [Affichage des rapports de test](#)
- [Autorisations relatives aux rapports de test](#)
- [Statuts des rapports de test](#)

Création de rapports de test

Pour créer un rapport de test, vous exécutez un projet de construction qui est configuré avec un à cinq groupes de rapports dans son fichier buildspec. Un rapport de test est créé pendant l'exécution. Il contient les résultats des cas de test spécifiés pour les groupes de rapports. Un nouveau rapport de test est généré pour chaque génération ultérieure qui utilise le même fichier buildspec.

Pour créer un rapport de test

1. Créez un projet de génération. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#).

2. Configurez le fichier buildspec de votre projet avec les informations de rapport de test :
 - a. Ajoutez une `reports :` section et spécifiez soit l'ARN d'un groupe de rapports existant, soit le nom d'un groupe de rapports.

Si vous spécifiez un ARN, CodeBuild utilise ce groupe de rapports.

Si vous spécifiez un nom, CodeBuild crée un groupe de rapports pour vous en utilisant le nom de votre projet, et le nom que vous avez spécifié, au format `<project-name>-<report-group-name>`. Si le groupe de rapports nommé existe déjà, CodeBuild utilise ce groupe de rapports.
 - b. Dans le groupe de rapports, spécifiez l'emplacement des fichiers contenant les résultats des tests. Si vous utilisez plusieurs groupes de rapports, spécifiez les emplacements des fichiers de résultats de test pour chacun d'eux. Un nouveau rapport de test est créé chaque fois que votre projet de génération s'exécute. Pour de plus amples informations, veuillez consulter [Spécification des fichiers de test](#).
 - c. Dans la section `commands` de la séquence `build` ou `post_build`, spécifiez les commandes qui exécutent les cas de test que vous avez spécifiés pour vos groupes de rapports. Pour de plus amples informations, veuillez consulter [Spécification des commandes de test](#).

Voici un exemple de section buildspec `reports` :

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Exécutez une version du projet de génération. Pour de plus amples informations, veuillez consulter [Exécuter AWS CodeBuild les builds manuellement](#).
4. Une fois la génération terminée, choisissez la nouvelle version dans Historique de génération sur la page de votre projet. Choisissez Rapports pour afficher le rapport de test. Pour de plus amples informations, veuillez consulter [Affichage des rapports de test d'une génération](#).

Création de rapports sur la couverture du code

CodeBuild vous permet de générer des rapports de couverture de code pour vos tests. Les rapports de couverture des codes suivants sont fournis :

Couverture de la ligne

La couverture linéaire mesure le nombre de déclarations couvertes par vos tests. Une instruction est une instruction unique, sans commentaires ni conditions.

```
line coverage = (total lines covered)/(total number of lines)
```

Couverture des succursales

La couverture des branches mesure le nombre de branches couvertes par vos tests parmi toutes les branches possibles d'une structure de contrôle, comme une case instruction `if` or.

```
branch coverage = (total branches covered)/(total number of branches)
```

Les formats de fichier de rapport de couverture de code suivants sont pris en charge :

- JaCoCo XML
- SimpleCov JSON¹
- Trèfle XML
- Couverture XML
- INFORMATIONS SUR LE LCOV

¹ [CodeBuild accepte les rapports de couverture du code JSON générés par simplecov, et non par simplecov-json.](#)

Création d'un rapport sur la couverture du code

Pour créer un rapport de couverture de code, vous exécutez un projet de génération configuré avec au moins un groupe de rapports de couverture de code dans son fichier `buildspec`. CodeBuild interprétera les résultats de couverture du code et fournira un rapport de couverture du code pour l'exécution. Un nouveau rapport de test est généré pour chaque génération ultérieure qui utilise le même fichier `buildspec`.

Pour créer un rapport de test

1. Créez un projet de génération. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#).
2. Configurez le fichier buildspec de votre projet avec les informations du rapport de test :
 - a. Ajoutez une `reports` : section et spécifiez le nom de votre groupe de rapports. CodeBuild crée un groupe de rapports pour vous en utilisant le nom de votre projet et le nom que vous avez spécifié au format `project-name-report-group-name-in-buildspec`. Si vous avez déjà un groupe de rapports à utiliser, spécifiez son ARN. Si vous utilisez le nom au lieu de l'ARN, CodeBuild crée un nouveau groupe de rapports. Pour de plus amples informations, veuillez consulter [Reports syntax in the buildspec file](#).
 - b. Dans le groupe de rapports, spécifiez l'emplacement des fichiers contenant les résultats de couverture du code. Si vous utilisez plusieurs groupes de rapports, spécifiez l'emplacement des fichiers de résultats pour chaque groupe de rapports. Un nouveau rapport de couverture du code est créé chaque fois que votre projet de construction s'exécute. Pour de plus amples informations, veuillez consulter [Spécification des fichiers de test](#).

Il s'agit d'un exemple qui génère un rapport de couverture de code pour un fichier de résultats JaCoCo XML situé dans `test-results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

- c. Dans la `commands` section de la `post_build` séquence `build or`, spécifiez les commandes qui exécutent l'analyse de couverture du code. Pour de plus amples informations, veuillez consulter [Spécification des commandes de test](#).
3. Exécutez une version du projet de génération. Pour de plus amples informations, veuillez consulter [Exécuter AWS CodeBuild les builds manuellement](#).
 4. Une fois la génération terminée, choisissez la nouvelle version dans Historique de génération sur la page de votre projet. Choisissez Rapports pour afficher le rapport sur la couverture du code. Pour de plus amples informations, veuillez consulter [Affichage des rapports de test d'une génération](#).

Découvrez automatiquement les rapports dans CodeBuild

La détection automatique permet de CodeBuild parcourir tous vos fichiers de compilation une fois la phase de génération terminée, de rechercher tous les types de fichiers de rapport pris en charge et de créer automatiquement de nouveaux groupes et rapports de test et de couverture de code. Pour tous les types de rapports découverts, CodeBuild crée de nouveaux groupes de rapports selon le modèle suivant :

```
<project-name>-<report-file-format>-AutoDiscovered
```

Note

Si les fichiers de rapport découverts ont le même type de format, ils seront placés dans le même groupe de rapports ou dans le même rapport.

La découverte automatique des rapports est configurée par les variables d'environnement de votre projet :

CODEBUILD_CONFIG_AUTO_DISCOVER

Cette variable détermine si la découverte automatique des rapports est désactivée pendant la génération. Par défaut, la découverte automatique des rapports est activée pour toutes les versions. Pour désactiver cette fonctionnalité, réglez CODEBUILD_CONFIG_AUTO_DISCOVER sur `false`.

CODEBUILD_CONFIG_AUTO_DISCOVER_DIR

(Facultatif) Cette variable détermine où sont CodeBuild recherchées les fichiers de rapport potentiels. Notez que par défaut, les CodeBuild recherches sont effectuées `**/*` par défaut.

Ces variables d'environnement peuvent être modifiées pendant la phase de construction. Par exemple, si vous souhaitez uniquement activer la découverte automatique des rapports pour les versions de la branche `main` git, vous pouvez vérifier la branche git pendant le processus de génération et CODEBUILD_CONFIG_AUTO_DISCOVER définir la valeur `false` si la version ne se trouve pas sur la `main` branche. La découverte automatique des rapports peut être désactivée à l'aide de la console ou des variables d'environnement du projet.

Rubriques

- [Configurer la découverte automatique des rapports à l'aide de la console](#)
- [Configurer la découverte automatique des rapports à l'aide des variables d'environnement du projet](#)

Configurer la découverte automatique des rapports à l'aide de la console

Utilisez la procédure suivante pour configurer la découverte automatique des rapports à l'aide de la console.

Pour configurer la découverte automatique des rapports à l'aide de la console

1. Créez un projet de construction ou choisissez un projet de construction à modifier. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) ou [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).
2. Dans Environnement, sélectionnez Configuration supplémentaire.
3. Pour désactiver la découverte automatique des rapports, dans la section Découverte automatique des rapports, sélectionnez Désactiver la découverte automatique des rapports.
4. (Facultatif) Dans Répertoire de découverte automatique (facultatif), entrez un modèle de répertoire pour rechercher les fichiers CodeBuild au format de rapport pris en charge. Notez que la CodeBuild recherche est `**/*` effectuée par défaut.

Configurer la découverte automatique des rapports à l'aide des variables d'environnement du projet

Utilisez la procédure suivante pour configurer la découverte automatique des rapports à l'aide des variables d'environnement du projet.

Pour configurer la découverte automatique des rapports à l'aide des variables d'environnement du projet

1. Créez un projet de construction ou choisissez un projet de construction à modifier. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) ou [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).
2. Dans Variables d'environnement, procédez comme suit :

- a. Pour désactiver la découverte automatique des rapports, saisissez Nom **CODEBUILD_CONFIG_AUTO_DISCOVER** et Valeur. **false** Cela désactive la découverte automatique des rapports.
- b. (Facultatif) Pour Nom, entrez **CODEBUILD_CONFIG_AUTO_DISCOVER_DIR** et pour Valeur, entrez le répertoire dans lequel vous CodeBuild devez rechercher les fichiers au format de rapport pris en charge. Par exemple, `output/*xml` recherche des `.xml` fichiers dans le `output` répertoire

Groupes de rapports

Un groupe de rapports contient des rapports de test et spécifie des paramètres partagés. Vous utilisez le fichier `buildspec` pour spécifier les cas de test à exécuter et les commandes pour les exécuter lors de la génération. Pour chaque groupe de rapports configuré dans un projet de génération, une exécution du projet de génération crée un rapport de test. Plusieurs exécutions d'un projet de génération configurées avec un groupe de rapports créent plusieurs rapports de test dans ce groupe de rapports, chacun avec les résultats des mêmes cas de test spécifiés pour ce groupe de rapports.

Les cas de test sont spécifiés pour un groupe de rapports dans le fichier `buildspec` d'un projet de génération. Vous pouvez spécifier jusqu'à cinq groupes de rapports dans un projet de génération. Lorsque vous exécutez une génération, tous les cas de test s'exécutent. Un nouveau rapport de test est créé avec les résultats de chaque cas de test spécifié pour un groupe de rapports. Chaque fois que vous exécutez une nouvelle version, les cas de test s'exécutent et un nouveau rapport de test est créé avec les nouveaux résultats de test.

Les groupes de rapports peuvent être utilisés dans plusieurs projets de génération. Tous les rapports de test créés avec un groupe de rapports partagent la même configuration, telle que son option d'exportation et ses autorisations, même si les rapports de test sont créés à l'aide de différents projets de génération. Les rapports de test créés avec un groupe de rapports dans plusieurs projets de génération peuvent contenir les résultats de l'exécution de différents ensembles de cas de test (un ensemble de cas de test pour chaque projet de génération). En effet, vous pouvez spécifier différents fichiers de cas de test pour le groupe de rapports dans le fichier `buildspec` de chaque projet. Vous pouvez également modifier les fichiers de cas de test d'un groupe de rapports dans un projet de génération en modifiant son fichier `buildspec`. Les versions ultérieures créent de nouveaux rapports de test qui contiennent les résultats des fichiers de cas de test dans le `buildspec` mis à jour.

Rubriques

- [Créer un groupe de rapports](#)
- [Attribution des noms des groupes de rapports](#)
- [Partager des groupes de rapports](#)
- [Spécification des fichiers de test](#)
- [Spécification des commandes de test](#)
- [Marquer un groupe de rapports dans AWS CodeBuild](#)
- [Mise à jour d'un groupe de rapports](#)

Créer un groupe de rapports

Vous pouvez utiliser la CodeBuild console AWS CLI, le fichier ou un fichier buildspec pour créer un groupe de rapports. Votre rôle IAM doit disposer des autorisations requises pour créer un groupe de rapports. Pour de plus amples informations, veuillez consulter [Autorisations relatives aux rapports de test](#).

Rubriques

- [Création d'un groupe de rapports \(buildspec\)](#)
- [Créer un groupe de rapports \(console\)](#)
- [Création d'un groupe de rapports \(CLI\)](#)
- [Création d'un groupe de rapports \(CloudFormation\)](#)

Création d'un groupe de rapports (buildspec)

Un groupe de rapports créé à l'aide de buildspec n'exporte pas les fichiers de résultats de test bruts. Vous pouvez afficher votre groupe de rapports et spécifier les paramètres d'exportation. Pour de plus amples informations, veuillez consulter [Mise à jour d'un groupe de rapports](#).

Pour créer un groupe de rapports à l'aide d'un fichier buildspec

1. Choisissez un nom de groupe de rapports qui n'est pas associé à un groupe de rapports dans votre AWS compte.
2. Configurez la section `reports` du fichier buildspec avec ce nom. Dans cet exemple, le nom du groupe de rapports est `new-report-group` et les cas de test d'utilisation sont créés avec le JUnit framework :

```
reports:
  new-report-group: #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Le nom du groupe de rapports peut également être spécifié à l'aide de variables d'environnement dans le buildspec :

```
version: 0.2
env:
  variables:
    REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
...
reports:
  $REPORT_GROUP_NAME:
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Pour plus d'informations, consultez [Spécification des fichiers de test](#) et [Reports syntax in the buildspec file](#).

3. Dans la section `commands`, spécifiez la commande pour exécuter vos tests. Pour de plus amples informations, veuillez consulter [Spécification des commandes de test](#).
4. Exécutez la génération. Une fois la génération terminée, un nouveau groupe de rapports est créé avec un nom qui utilise le format `project-name-report-group-name`. Pour de plus amples informations, veuillez consulter [Attribution des noms des groupes de rapports](#).

Créer un groupe de rapports (console)

Utilisez la procédure suivante pour créer un groupe de rapports à l'aide du AWS Management Console.

Pour créer un groupe de rapports

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez Créer un groupe de rapports.
4. Pour Nom du groupe de rapports, entrez un nom pour votre groupe de rapports.
5. (Facultatif) Pour Tags, entrez le nom et la valeur de tous les tags que vous souhaitez que les AWS services d'assistance utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.
6. Si vous souhaitez télécharger les données brutes des résultats de votre rapport de test dans un compartiment Amazon S3 :
 - a. Sélectionnez Exporter vers Amazon S3.
 - b. Pour le nom du compartiment S3, entrez le nom du compartiment S3.
 - c. (Facultatif) Pour le propriétaire du AWS compartiment S3, entrez l'identifiant du compte propriétaire du compartiment S3. Cela permet de signaler les données à exporter vers un compartiment Amazon S3 appartenant à un compte autre que celui exécutant la build.
 - d. Pour Préfixe du chemin d'accès, entrez le chemin de votre compartiment S3 où vous souhaitez télécharger les résultats de vos tests.
 - e. Sélectionnez Compresser les données de résultats de test dans un fichier zip pour compresser vos fichiers de données de résultats de test bruts.
 - f. Développez Configuration supplémentaire pour afficher les options de chiffrement. Sélectionnez l'une des méthodes suivantes :
 - Clé AWS gérée par défaut à utiliser Clé gérée par AWS pour Amazon S3. Pour plus d'informations, consultez la section [Gestion des clients CMKs](#) dans le guide de AWS Key Management Service l'utilisateur. Il s'agit de l'option de chiffrement par défaut.
 - Choisissez une clé personnalisée pour utiliser une clé gérée par le client que vous créez et configurez. Pour la clé de AWS KMS chiffrement, entrez l'ARN de votre clé de chiffrement. Son format est `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Pour de plus amples informations, veuillez consulter [Création de clés KMS](#) dans le Guide de l'utilisateur AWS Key Management Service .
 - Désactivez le chiffrement des artefacts pour désactiver le chiffrement. Vous pouvez choisir cette option si vous souhaitez partager vos résultats de test ou les publier sur

un site web statique. (Un site web dynamique peut exécuter du code pour déchiffrer les résultats des tests.)

Pour de plus amples informations sur le chiffrement au repos, veuillez consulter [Chiffrement des données](#).

 Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

7. Choisissez Créer un groupe de rapports.

Création d'un groupe de rapports (CLI)

Utilisez la procédure suivante pour créer un groupe de rapports à l'aide du AWS CLI.

Pour créer un groupe de rapports

1. Créez un fichier nommé `CreateReportGroup.json`.
2. En fonction de vos besoins, copiez l'un des extraits de code JSON suivants dans `CreateReportGroup.json` :
 - Utilisez le JSON suivant pour spécifier que votre groupe de rapports de test exporte les fichiers de résultats de test bruts vers un compartiment Amazon S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    }
  },
}
```

```
"tags": [  
  {  
    "key": "tag-key",  
    "value": "tag-value"  
  }  
]  
}  
}
```

- **<bucket-name>** Remplacez-le par le nom de votre compartiment Amazon S3 et **<path>** par le chemin de votre compartiment vers l'endroit où vous souhaitez exporter les fichiers.
- Si vous souhaitez compresser les fichiers exportés, pour packaging, spécifiez ZIP. Sinon, spécifiez NONE.
- bucketOwner est facultatif et n'est obligatoire que si le compartiment Amazon S3 appartient à un compte autre que celui qui exécute le build.
- Utilisez encryptionDisabled de spécifier si les fichiers exportés doivent être chiffrés. Si vous cryptez les fichiers exportés, entrez votre clé gérée par le client. Pour de plus amples informations, veuillez consulter [Mise à jour d'un groupe de rapports](#).
- Utilisez le JSON suivant pour spécifier que votre rapport de test n'exporte pas les fichiers de test bruts :

```
{  
  "name": "<report-name>",  
  "type": "TEST",  
  "exportConfig": {  
    "exportConfigType": "NO_EXPORT"  
  }  
}
```

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

3. Exécutez la commande suivante :

```
aws codebuild create-report-group --cli-input-json file://  
CreateReportGroupInput.json
```

Création d'un groupe de rapports (CloudFormation)

Utilisez les instructions suivantes pour créer un groupe de rapports à l'aide du CloudFormation modèle

Pour créer un groupe de rapports à l'aide du CloudFormation modèle

Vous pouvez utiliser un fichier CloudFormation modèle pour créer et approvisionner un groupe de rapports. Pour plus d'informations, consultez le [CloudFormation guide de l'utilisateur](#).

Le modèle CloudFormation YAML suivant crée un groupe de rapports qui n'exporte pas les fichiers de résultats de test bruts.

```
Resources:  
  CodeBuildReportGroup:  
    Type: AWS::CodeBuild::ReportGroup  
    Properties:  
      Name: my-report-group-name  
      Type: TEST  
      ExportConfig:  
        ExportConfigType: NO_EXPORT
```

Le modèle CloudFormation YAML suivant crée un groupe de rapports qui exporte les fichiers de résultats de test bruts vers un compartiment Amazon S3.

```
Resources:  
  CodeBuildReportGroup:  
    Type: AWS::CodeBuild::ReportGroup  
    Properties:  
      Name: my-report-group-name  
      Type: TEST  
      ExportConfig:  
        ExportConfigType: S3  
        S3Destination:  
          Bucket: amzn-s3-demo-bucket  
          Path: path-to-folder-for-exported-files  
          Packaging: ZIP
```

```
EncryptionKey: my-KMS-encryption-key  
EncryptionDisabled: false
```

 Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Attribution des noms des groupes de rapports

Lorsque vous utilisez la console AWS CLI ou la AWS CodeBuild console pour créer un groupe de rapports, vous spécifiez un nom pour le groupe de rapports. Si vous utilisez la spécification de build pour créer un nouveau groupe de rapports, il est nommé en utilisant le format *project-name-report-group-name-specified-in-buildspec*. Tous les rapports créés en exécutant des versions de ce projet de génération appartiennent au nouveau groupe de rapports portant le nouveau nom.

Si vous ne souhaitez pas CodeBuild créer de nouveau groupe de rapports, spécifiez l'ARN du groupe de rapports dans le fichier buildspec d'un projet de construction. Vous pouvez spécifier l'ARN d'un groupe de rapports dans plusieurs projets de génération. Après l'exécution de chaque projet de génération, le groupe de rapports contient des rapports de test créés par chaque projet de génération.

Par exemple, si vous créez un groupe de rapports portant le nom `my-report-group`, puis que vous utilisez son nom dans deux projets de génération différents nommés `my-project-1` et `my-project-2` et que vous créez une version des deux projets, deux nouveaux groupes de rapports sont créés. Le résultat est trois groupes de rapports portant les noms suivants :

- `my-report-group` : n'a pas de rapport de test.
- `my-project-1-my-report-group` : contient des rapports avec les résultats des tests exécutés par le projet de génération nommé `my-project-1`.
- `my-project-2-my-report-group` : contient des rapports avec les résultats des tests exécutés par le projet de génération nommé `my-project-2`.

Si vous utilisez l'ARN du groupe de rapports nommé `my-report-group` dans les deux projets, puis que vous exécutez des versions de chaque projet, vous disposez toujours d'un groupe de rapports

(my-report-group). Ce groupe de rapports contient des rapports de test avec les résultats des tests exécutés par les deux projets de génération.

Si vous choisissez un nom de groupe de rapports qui n'appartient pas à un groupe de rapports dans votre compte AWS, puis que vous utilisez ce nom pour un groupe de rapports dans un fichier buildspec et exécutez une génération de son projet de génération, un nouveau groupe de rapports est créé. Le format du nom du nouveau groupe de rapports est *project-name-new-group-name*. Par exemple, si aucun groupe de rapports ne porte ce nom new-report-group dans votre AWS compte et que vous le spécifiez dans un projet de génération appelé test-project, une exécution de génération crée un nouveau groupe de rapports portant ce nom test-project-new-report-group.

Partager des groupes de rapports

Le partage de groupes de rapports permet à plusieurs AWS comptes ou utilisateurs de consulter un groupe de rapports, ses rapports non expirés et les résultats des tests de ses rapports. Dans ce modèle, le compte propriétaire du groupe de rapports (propriétaire) partage un groupe de rapports avec d'autres comptes (consommateurs). Un consommateur ne peut pas modifier un groupe de rapports. Un rapport expire 30 jours après sa création.

Rubriques

- [Partager un groupe de rapports](#)
- [Services connexes](#)
- [Accédez aux groupes de rapports partagés avec vous](#)
- [Annuler le partage d'un groupe de rapports partagé](#)
- [Identifier un groupe de rapports partagé](#)
- [Autorisations de groupe de rapports partagés](#)

Partager un groupe de rapports

Lorsque vous partagez un groupe de rapports, le consommateur dispose d'un accès en lecture seule au groupe de rapports et à ses rapports. Le consommateur peut utiliser le AWS CLI pour consulter le groupe de rapports, ses rapports et les résultats des scénarios de test pour chaque rapport. Le consommateur ne peut pas :

- Consulter un groupe de rapports partagé ou ses rapports dans la CodeBuild console.
- Modifier un groupe de rapports partagé.

- Utiliser l'ARN du groupe de rapports partagé dans un projet pour exécuter un rapport. Une génération de projet qui spécifie un groupe de rapports partagé échoue.

Vous pouvez utiliser la CodeBuild console pour ajouter un groupe de rapports à un partage de ressources existant. Si vous souhaitez ajouter le groupe de rapports à un nouveau partage de ressources, vous devez d'abord le créer dans la [console AWS RAM](#).

Pour partager un groupe de rapports avec des unités organisationnelles ou une organisation entière, vous devez activer le partage avec AWS Organizations. Pour de plus amples informations, veuillez consulter [Activer le partage avec AWS Organizations](#) dans le Guide de l'utilisateur AWS RAM .

Vous pouvez utiliser la CodeBuild console, la AWS RAM console ou AWS CLI pour partager des groupes de rapports dont vous êtes propriétaire.

Prérequis

Pour partager un groupe de rapports, votre AWS compte doit en être le propriétaire. Vous ne pouvez pas partager un groupe de rapports qui a été partagé avec vous.

Pour partager un groupe de rapports dont vous êtes le propriétaire (CodeBuild console)

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le projet que vous souhaitez partager, puis choisissez Partager. Pour de plus amples informations, veuillez consulter [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM .

Pour partager des groupes de rapports dont vous êtes propriétaire (AWS RAM console)

Consultez [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM .

Pour partager des groupes de rapports dont vous êtes propriétaire (AWS RAM commande)

Utilisez la commande [create-resource-share](#).

Pour partager un groupe de rapports dont vous êtes le propriétaire (CodeBuild commande)

Utilisez la commande [put-resource-policy](#) :

1. Créez un fichier nommé `policy.json` et copiez ce qui suit dans celui-ci.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"],
    "Resource": "arn:aws:iam::*:role/Service*"
  ]
}
```

2. Mettez à jour `policy.json` avec l'ARN du groupe de rapports et les identificateurs avec lesquels vous le partagez. L'exemple suivant accorde un accès en lecture seule au groupe de rapports avec l'ARN `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` à Alice et à l'utilisateur root pour le AWS compte identifié par 123456789012.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::123456789012:user/Alice",
        "123456789012"
      ]
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",

```

```
"codebuild:ListReportsForReportGroup",
"codebuild:DescribeTestCases"],
"Resource": "arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group"
}]
}
```

3. Exécutez la commande suivante.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://policy.json
```

Services connexes

Le partage de groupes de rapports s'intègre à AWS Resource Access Manager (AWS RAM), un service qui vous permet de partager vos AWS ressources avec n'importe quel AWS compte ou via AWS Organizations. Avec AWS RAM, vous partagez les ressources que vous possédez en créant un partage de ressources qui spécifie les ressources et les consommateurs avec lesquels les partager. Les consommateurs peuvent être AWS des comptes individuels AWS Organizations, des unités organisationnelles ou une organisation entière AWS Organizations.

Pour plus d'informations, consultez le Guide de l'utilisateur [AWS RAM](#).

Accédez aux groupes de rapports partagés avec vous

Pour accéder à un groupe de rapports partagé, le rôle IAM d'un consommateur requiert l'autorisation `BatchGetReportGroups`. Vous pouvez associer la politique suivante à leur rôle IAM :

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetReportGroups"
  ]
}
```

Pour de plus amples informations, veuillez consulter [Utilisation de politiques basées sur l'identité pour AWS CodeBuild](#).

Annuler le partage d'un groupe de rapports partagé

Un groupe de rapports non partagé, y compris ses rapports et leurs résultats de cas de test, est accessible uniquement par son propriétaire. Si vous résiliez le partage d'un groupe de rapports, les AWS comptes ou utilisateurs avec lesquels vous l'avez précédemment partagé ne peuvent pas accéder au groupe de rapports, à ses rapports ou aux résultats des tests figurant dans les rapports.

Pour annuler le partage d'un groupe de rapports partagé qui vous appartient, vous devez le supprimer du partage de ressources. Vous pouvez utiliser la AWS RAM console ou AWS CLI effectuer cette opération.

Pour annuler le partage d'un groupe de rapports partagé dont vous êtes le propriétaire (AWS RAM console)

Consultez la section [Mise à jour d'un partage de ressources](#) du Guide de l'utilisateur AWS RAM .

Pour annuler le partage d'un groupe de rapports partagé dont vous êtes le propriétaire (AWS RAM commande)

Utilisez la commande [disassociate-resource-share](#).

Pour annuler le partage du groupe de rapports dont vous êtes le propriétaire (CodeBuild commande)

Exécutez la [delete-resource-policy](#) commande et spécifiez l'ARN du groupe de rapports dont vous souhaitez annuler le partage :

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Identifier un groupe de rapports partagé

Les propriétaires et les consommateurs peuvent utiliser le AWS CLI pour identifier les groupes de rapports partagés.

Pour identifier et obtenir des informations sur un groupe de rapports partagé et ses rapports, utilisez les commandes suivantes :

- Pour voir les groupes ARNs de rapports partagés avec vous, exécutez [list-shared-report-groups](#) :

```
aws codebuild list-shared-report-groups
```

- Pour voir les rapports ARNs d'un groupe de rapports, exécutez-le à [list-reports-for-report-group](#) à l'aide de l'ARN du groupe de rapports :

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Pour afficher des informations sur les cas de test dans un rapport, exécutez [describe-test-cases](#) à l'aide de l'ARN de rapport :

```
aws codebuild describe-test-cases --report-arn report-arn
```

Le résultat se présente comme suit :

```
{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "report-arn",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "path-to-output-report-files"
    },
    {
      "status": "SUCCEEDED",
      "name": "Test case 2",
      "expired": 1575916770.0,
      "reportArn": "report-arn",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "path-to-output-report-files"
    }
  ]
}
```

Autorisations de groupe de rapports partagés

Autorisations accordées aux propriétaires

Un propriétaire de groupe de rapports peut modifier le groupe de rapports et le spécifier dans un projet pour exécuter des rapports.

Autorisations accordées aux consommateurs

Un consommateur de groupe de rapports peut afficher un groupe de rapports, ses rapports et les résultats de cas de test pour ses rapports. Un consommateur ne peut pas modifier un groupe de rapports ou ses rapports, et ne peut pas l'utiliser pour créer des rapports.

Spécification des fichiers de test

Vous spécifiez les fichiers de résultats de test et leur emplacement pour chaque groupe de rapports dans la section `reports` du fichier `buildspec` de votre projet de génération. Pour de plus amples informations, veuillez consulter [Reports syntax in the buildspec file](#).

Voici un exemple de section `reports` qui spécifie deux groupes de rapports pour un projet de génération. L'un est spécifié avec son ARN, l'autre avec un nom. La section `files` spécifie les fichiers qui contiennent les résultats de cas de test. La section facultative `base-directory` spécifie le répertoire où se trouvent les fichiers de cas de test. La section facultative `discard-paths` indique si les chemins d'accès aux fichiers de résultats de test chargés dans un compartiment Amazon S3 sont supprimés.

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  #surefire junit reports
  files:
    - '**/*'
  base-directory: 'surefire/target/surefire-reports'
  discard-paths: false

sampleReportGroup: #Cucumber reports from json plugin
  files:
    - 'cucumber-json/target/cucumber-json-report.json'
  file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Spécification des commandes de test

Vous spécifiez les commandes qui exécutent vos cas de test dans la section `commands` de votre fichier `buildspec`. Ces commandes exécutent les cas de test spécifiés pour vos groupes de rapports dans la section `reports` de votre fichier `buildspec`. Voici un exemple de section `commands` qui inclut des commandes pour exécuter les tests dans les fichiers de test :

```
commands:
  - echo Running tests for surefire junit
  - mvn test -f surefire/pom.xml -fn
  - echo
  - echo Running tests for cucumber with json plugin
  - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
    cucumber-json/pom.xml -fn
```

Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

Marquer un groupe de rapports dans AWS CodeBuild

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou AWS assignez à une AWS ressource. Chaque AWS étiquette comporte deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment`, `Project` ou `Secret`). Les touches de tag distinguent les majuscules et minuscules.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333`, `Production` ou le nom d'une équipe). Omettre la valeur de balise équivaut à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Ensemble, ces éléments sont connus sous le nom de paires clé-valeur. Pour les limites sur le nombre de balises que vous pouvez avoir dans un groupe de rapports et les restrictions sur les valeurs et clés de balise, consultez [Étiquettes](#).

Les balises vous aident à identifier et à organiser vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à un groupe de CodeBuild rapports que celle que vous attribuez à un compartiment Amazon S3. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc sur les [bonnes pratiques de balisage](#).

Dans CodeBuild, les ressources principales sont le groupe de rapports et le projet. Vous pouvez utiliser la CodeBuild console, le AWS CLI CodeBuild APIs, ou AWS SDKs pour ajouter, gérer et supprimer des balises pour un groupe de rapports. Outre l'identification, l'organisation et le suivi de votre groupe de rapports à l'aide de balises, vous pouvez utiliser des balises dans les politiques IAM pour contrôler qui peut consulter votre groupe de rapports et interagir avec celui-ci. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Rubriques

- [Ajouter des balises à un groupe de rapports](#)
- [Afficher les balises d'un groupe de rapports](#)
- [Modifier les balises d'un groupe de rapports](#)
- [Supprimer des balises d'un groupe de rapports](#)

Ajouter des balises à un groupe de rapports

L'ajout de balises à un groupe de rapports peut vous aider à identifier et à organiser vos AWS ressources et à gérer l'accès à celles-ci. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à un groupe de rapports. N'oubliez pas qu'il y a des limites sur le nombre de balises que vous pouvez avoir dans un groupe de rapports. Il existe des restrictions sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour de plus amples informations, veuillez consulter [Étiquettes](#). Une fois que vous avez des balises, vous pouvez créer des politiques IAM pour gérer l'accès au groupe de rapports en fonction de ces balises. Vous pouvez utiliser la CodeBuild console ou le AWS CLI pour ajouter des balises à un groupe de rapports.

Important

L'ajout de balises à un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant d'ajouter une balise à un groupe de rapports, assurez-vous de consulter les politiques IAM susceptibles d'utiliser des balises pour contrôler l'accès aux ressources telles que les groupes de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Pour plus d'informations sur l'ajout de balises à un groupe de rapports lorsque vous le créez, consultez [Créer un groupe de rapports \(console\)](#).

Rubriques

- [Ajouter une balise à un groupe de rapports \(console\)](#)
- [Ajouter une balise à un groupe de rapports \(AWS CLI\)](#)

Ajouter une balise à un groupe de rapports (console)

Vous pouvez utiliser la CodeBuild console pour ajouter une ou plusieurs balises à un groupe de CodeBuild rapports.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Report groups (Groupes de rapports), choisissez le nom du groupe de rapports auquel vous souhaitez ajouter des balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Si aucune des balises n'a été ajoutée au groupe de rapports, choisissez Add tag (Ajouter une balise). Vous pouvez également choisir Modifier, puis Ajouter une balise.
5. Dans Key (Clé), entrez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).
6. (Facultatif) Pour ajouter une autre balise, choisissez à nouveau Add tag (Ajouter une balise).
7. Lorsque vous avez fini d'ajouter des balises, choisissez Submit (Envoyer).

Ajouter une balise à un groupe de rapports (AWS CLI)

Pour ajouter une balise à un groupe de rapports lorsque vous le créez, veuillez consulter [Création d'un groupe de rapports \(CLI\)](#). Dans `CreateReportGroup.json`, ajoutez vos balises.

Pour ajouter des balises à un groupe de rapports existant, veuillez consulter [Mise à jour d'un groupe de rapports \(CLI\)](#) et ajoutez vos balises dans `UpdateReportGroupInput.json`.

Dans ces étapes, nous supposons que vous avez déjà installé une version récente de l' AWS CLI ou que vous avez procédé à une mise à jour vers la version actuelle. Pour plus d'informations, consultez [Installing the AWS Command Line Interface](#) (Installation de).

Afficher les balises d'un groupe de rapports

Les balises peuvent vous aider à identifier et à organiser vos AWS ressources et à gérer l'accès à celles-ci. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc

sur les [bonnes pratiques de balisage](#). Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Deny or allow actions on report groups based on resource tags](#).

Afficher les balises d'un groupe de rapports (console)

Vous pouvez utiliser la CodeBuild console pour afficher les balises associées à un groupe de CodeBuild rapports.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dans lequel vous souhaitez afficher les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).

Afficher les balises d'un groupe de rapports (AWS CLI)

Procédez comme suit pour utiliser le AWS CLI afin d'afficher les AWS balises d'un groupe de rapports. Si aucune balise n'a été ajoutée, la liste des balises renvoyées est vide.

1. Utilisez la console ou le AWS CLI pour localiser l'ARN de votre groupe de rapports. Prenez-en note.

AWS CLI

Exécutez la commande suivante.

```
aws list-report-groups
```

Cette commande renvoie des informations au format JSON semblables à ce qui suit :

```
{
  "reportGroups": [
    "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
  ]
}
```

Un ARN de groupe de rapports se termine par son nom, que vous pouvez utiliser pour identifier l'ARN de votre groupe de rapports.

Console

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
 2. Dans Groupes de rapports, choisissez le nom de votre groupe de rapports avec les balises que vous souhaitez afficher.
 3. Dans Configuration recherchez l'ARN de votre groupe de rapports.
2. Exécutez la commande suivante. Utilisez l'ARN dont vous avez fait une note pour le paramètre `--report-group-arns`.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

En cas de succès, cette commande renvoie des informations au format JSON qui contiennent une section `tags` similaire à la suivante :

```
{
  ...
  "tags": {
    "Status": "Secret",
    "Project": "TestBuild"
  }
  ...
}
```

Modifier les balises d'un groupe de rapports

Vous pouvez modifier la valeur d'une balise associée à un groupe de rapports. Vous pouvez également modifier le nom de la clé, ce qui équivaut à supprimer la balise et à ajoutant une carte différente avec le nouveau nom et la même valeur que l'autre clé. N'oubliez pas qu'il y a des limites sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour de plus amples informations, veuillez consulter [Étiquettes](#).

Important

La modification des balises d'un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant de modifier le nom (clé) ou la valeur d'une balise pour un groupe de rapports, assurez-vous de consulter les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources telles que les groupes de rapports.

Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Deny or allow actions on report groups based on resource tags](#).

Modifier une balise pour un groupe de rapports (console)

Vous pouvez utiliser la CodeBuild console pour modifier les balises associées à un groupe de CodeBuild rapports.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dont vous souhaitez modifier les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Choisissez Modifier.
5. Effectuez l'une des actions suivantes :
 - Pour modifier la balise, entrez un nouveau nom dans Key (Clé). La modification du nom de la balise est l'équivalent de la suppression d'une balise et de l'ajout d'une nouvelle balise avec le nouveau nom de clé.
 - Pour modifier la valeur d'une balise, saisissez une nouvelle valeur. Si vous souhaitez modifier la valeur en valeur nulle, supprimez la valeur actuelle et laissez le champ vide.
6. Lorsque vous avez terminé de modifier des balises, choisissez Submit (Soumettre).

Modifier les balises d'un groupe de rapports (AWS CLI)

Pour ajouter, modifier ou supprimer des balises d'un groupe de rapports, veuillez consulter [Mise à jour d'un groupe de rapports \(CLI\)](#). Mettez à jour les balises dans `UpdateReportGroupInput.json`.

Supprimer des balises d'un groupe de rapports

Vous pouvez supprimer une ou plusieurs balises associées à un groupe de rapports. La suppression d'un tag ne supprime pas le tag des autres AWS ressources associées à ce tag.

Important

La suppression de balises pour un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant de supprimer une balise d'un groupe de rapports, assurez-vous

de passer en revue les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources telles que les groupes de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources](#).

Supprimer une balise d'un groupe de rapports (console)

Vous pouvez utiliser la CodeBuild console pour supprimer l'association entre une balise et un groupe de CodeBuild rapports.

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dans lequel vous souhaitez supprimer les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Choisissez Modifier.
5. Trouvez la balise que vous voulez supprimer, puis choisissez Remove tag (Supprimer une balise).
6. Lorsque vous avez terminé de supprimer les balises, choisissez Submit (Envoyer).

Supprimer une balise d'un groupe de rapports (AWS CLI)

Procédez comme suit pour utiliser le AWS CLI pour supprimer une balise d'un groupe de CodeBuild rapports. La suppression d'une balise supprime uniquement son association au groupe de rapports, mais pas la balise en elle-même.

Note

Si vous supprimez un groupe de CodeBuild rapports, toutes les associations de balises sont supprimées du groupe de rapports supprimé. Vous n'avez pas besoin de supprimer les balises avant de supprimer un groupe de rapports.

Pour supprimer une ou plusieurs balises d'un groupe de rapports, reportez-vous à la section [Modifier les balises d'un groupe de rapports \(AWS CLI\)](#). Mettez à jour la tags section dans les données formatées JSON avec une liste mise à jour de balises qui ne contient pas celles que vous souhaitez supprimer. Si vous souhaitez supprimer toutes les balises, mettez à jour la section tags pour :

```
"tags: []"
```

Mise à jour d'un groupe de rapports

Lorsque vous mettez à jour un groupe de rapports, vous pouvez indiquer s'il convient d'exporter les données brutes des résultats de test vers des fichiers d'un compartiment Amazon S3. Si vous choisissez d'exporter vers un compartiment S3, vous pouvez spécifier les éléments suivants pour votre groupe de rapports :

- Indique si les fichiers de résultats de test bruts sont compressés dans un fichier ZIP.
- Indique si les fichiers de résultats de test bruts sont chiffrés. Vous pouvez spécifier le chiffrement avec l'une des options suivantes :
 - Et Clé gérée par AWS pour Amazon S3.
 - Une clé gérée par le client que vous créez et configurez.

Pour de plus amples informations, veuillez consulter [Chiffrement des données](#).

Si vous utilisez le AWS CLI pour mettre à jour un groupe de rapports, vous pouvez également mettre à jour ou ajouter des balises. Pour de plus amples informations, veuillez consulter [Marquer un groupe de rapports dans AWS CodeBuild](#).

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Rubriques

- [Mise à jour d'un groupe de rapports \(console\)](#)
- [Mise à jour d'un groupe de rapports \(CLI\)](#)

Mise à jour d'un groupe de rapports (console)

Utilisez la procédure suivante pour mettre à jour un groupe de rapports à l'aide du AWS Management Console.

Pour mettre à jour un groupe de rapports

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le groupe de rapports que vous souhaitez mettre à jour.
4. Choisissez Modifier.
5. Sélectionnez ou désactivez Backup to Amazon S3. Si vous avez sélectionné cette option, spécifiez vos paramètres d'exportation :
 - a. Pour le nom du compartiment S3, entrez le nom du compartiment S3.
 - b. Pour Préfixe du chemin d'accès, entrez le chemin de votre compartiment S3 où vous souhaitez télécharger les résultats de vos tests.
 - c. Sélectionnez Compresser les données de résultats de test dans un fichier zip pour compresser vos fichiers de données de résultats de test bruts.
 - d. Développez Configuration supplémentaire pour afficher les options de chiffrement. Sélectionnez l'une des méthodes suivantes :
 - Clé AWS gérée par défaut à utiliser Clé gérée par AWS pour Amazon S3. Pour plus d'informations, consultez la section [Gestion des clients CMKs](#) dans le guide de AWS Key Management Service l'utilisateur. Il s'agit de l'option de chiffrement par défaut.
 - Choisissez une clé personnalisée pour utiliser une clé gérée par le client que vous créez et configurez. Pour la clé de AWS KMS chiffrement, entrez l'ARN de votre clé de chiffrement. Son format est `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Pour de plus amples informations, veuillez consulter [Création de clés KMS](#) dans le Guide de l'utilisateur AWS Key Management Service .
 - Désactivez le chiffrement des artefacts pour désactiver le chiffrement. Vous pouvez choisir cette option si vous souhaitez partager vos résultats de test ou les publier sur un site web statique. (Un site web dynamique peut exécuter du code pour déchiffrer les résultats des tests.)

Mise à jour d'un groupe de rapports (CLI)

Utilisez la procédure suivante pour mettre à jour un groupe de rapports à l'aide du AWS CLI.

Pour mettre à jour un groupe de rapports

1. Créez un fichier nommé `UpdateReportGroupInput.json`.
2. Copiez ce qui suit dans `UpdateReportGroupInput.json` :

```
{
  "arn": "",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "bucket-name",
      "path": "path",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "your-key"
    }
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```

3. Entrez l'ARN de votre groupe de rapports dans la ligne `arn` (par exemple, `"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1"`)).
4. Mettez à jour `UpdateReportGroupInput.json` avec les mises à jour que vous souhaitez appliquer à votre groupe de rapports.
 - Si vous souhaitez mettre à jour votre groupe de rapports pour exporter les fichiers de résultats de test bruts vers un compartiment S3, mettez à jour la section `exportConfig`. Remplacez `bucket-name` par le nom de votre compartiment S3 et `path` par le chemin d'accès de votre compartiment S3 vers l'endroit où vous souhaitez exporter les fichiers. Si vous souhaitez compresser les fichiers exportés, pour `packaging`, spécifiez `ZIP`. Sinon, spécifiez `NONE`. Utilisez `encryptionDisabled` de spécifier si les fichiers exportés doivent être chiffrés. Si vous cryptez les fichiers exportés, entrez votre clé gérée par le client.

- Si vous souhaitez mettre à jour votre groupe de rapports afin qu'il n'exporte pas les fichiers de résultats de test bruts vers un compartiment S3, mettez à jour la section `exportConfig` avec le code JSON suivant :

```
{
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

- Si vous souhaitez mettre à jour les balises du groupe de rapports, mettez à jour la section `tags`. Vous pouvez modifier, ajouter ou supprimer des balises. Si vous voulez supprimer toutes les balises, mettez-le à jour avec le JSON suivant :

```
"tags": []
```

5. Exécutez la commande suivante :

```
aws codebuild update-report-group \
--cli-input-json file:///UpdateReportGroupInput.json
```

Cadres de test

Les rubriques de cette section montrent comment configurer les rapports de test dans AWS CodeBuild différents frameworks de test.

Rubriques

- [Configurer les rapports de test avec Jasmine](#)
- [Configurer les rapports de test avec Jest](#)
- [Configurer les rapports de test avec pytest](#)
- [Configurez les rapports de test avec RSpec](#)

Configurer les rapports de test avec Jasmine

La procédure suivante explique comment configurer les rapports de test dans le framework AWS CodeBuild de test [JasmineBDD](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un CodeBuild projet existant.
- Votre projet est un projet Node.js qui est configuré pour utiliser le cadre de test Jasmine.

Ajoutez le package [jasmine-reporters](#) à la section devDependencies du fichier `package.json` de votre projet. Ce paquet contient une collection de classes de JavaScript reporter qui peuvent être utilisées avec Jasmine.

```
npm install --save-dev jasmine-reporters
```

S'il n'est pas déjà présent, ajoutez le script `test` au fichier `package.json` de votre projet. Le script garantit que Jasmine est appelée lors de `npm test` son exécution.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild prend en charge les reporters de test Jasmine suivants :

JUnitXmlReporter

Permet de générer des rapports au format JUnitXML.

NUnitXmlReporter

Permet de générer des rapports au format NUnitXML.

Un projet Node.js avec Jasmine aura, par défaut, un sous-répertoire `spec`, qui contient la configuration de Jasmine et les scripts de test.

Pour configurer Jasmine pour générer des rapports au JUnitXML format, instanciez le `JUnitXmlReporter` rapporteur en ajoutant le code suivant à vos tests.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
```

```
    consolidateAll: true
  });

jasmine.getEnv().addReporter(junitReporter);
```

Pour configurer Jasmine pour générer des rapports au NunitXML format, instanciez le `NUnitXmlReporter` rapporteur en ajoutant le code suivant à vos tests.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(nunitReporter)
```

Les rapports de test sont exportés vers le fichier spécifié par `<test report directory>/<report filename>`.

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jasmine_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Si vous utilisez le format de rapport `NunitXml`, modifiez la valeur `file-format` comme suit.


```
file-format: NUNITXML
```

Configurer les rapports de test avec Jest

La procédure suivante explique comment configurer les rapports de test dans le [framework AWS CodeBuild de test Jest](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un CodeBuild projet existant.
- Votre projet est un projet Node.js qui est configuré pour utiliser le cadre de test Jest.

Ajoutez le [jest-junit](#) package à la devDependencies section du package.json fichier de votre projet. CodeBuild utilise ce package pour générer des rapports au JunitXml format.

```
npm install --save-dev jest-junit
```

S'il n'est pas déjà présent, ajoutez le script test au fichier package.json de votre projet. Le test script garantit que Jest est appelé lors npm test de son exécution.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Configurez Jest pour utiliser le reporter JunitXml en ajoutant ce qui suit à votre fichier de configuration Jest. Si votre projet ne possède pas de fichier de configuration Jest, créez un fichier nommé `jest.config.js` à la racine de votre projet et ajoutez ce qui suit. Les rapports de test sont exportés vers le fichier spécifié par `<test report directory>/<report filename>`.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
  ]
}
```

```
};
```

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jest_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Configurer les rapports de test avec pytest

La procédure suivante montre comment configurer les rapports de test dans le [framework AWS CodeBuild de test pytest](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un CodeBuild projet existant.
- Votre projet est un projet Python qui est configuré pour utiliser le cadre de test pytest.

Ajoutez l'entrée suivante à la phase `build` ou `post_build` de votre fichier `buildspec.yml`. Ce code découvre automatiquement les tests dans le répertoire en cours et exporte les rapports de test vers le fichier spécifié par `<test report directory>/<report filename>`. Le rapport utilise le format `JUnitXml`.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Configurez les rapports de test avec RSpec

La procédure suivante explique comment configurer les rapports de test dans AWS CodeBuild le [cadre de RSpec test](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un CodeBuild projet existant.
- Votre projet est un projet Ruby configuré pour utiliser le framework de RSpec test.

Ajoutez/mettez à jour ce qui suit dans votre fichier `buildspec.yml`. Ce code exécute les tests dans le `<test source directory>` répertoire et exporte les rapports de test vers le fichier spécifié par `<test report directory>/<report filename>`. Le rapport utilise le format `JUnitXml`.

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  pre_build:
    commands:
```

```
- gem install rspec
- gem install rspec_junit_formatter
build:
  commands:
    - rspec <test source directory>/* --format RspecJunitFormatter --out <test report
      directory>/<report filename>
reports:
  rspec_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Affichage des rapports de test

Vous pouvez afficher des détails sur un rapport de test, tels que des informations sur ses cas de test, ses nombres de réussite et d'échec, et la durée de son exécution. Vous pouvez consulter les rapports de test regroupés par build, par groupe de rapports ou par AWS compte. Choisissez un rapport de test dans la console pour afficher les détails et les résultats de ses scénarios de test.

Vous pouvez afficher les rapports de test qui n'ont pas expiré. Les rapports de test expirent 30 jours après leur création. Vous ne pouvez pas afficher un rapport expiré dans CodeBuild.

Rubriques

- [Affichage des rapports de test d'une génération](#)
- [Affichage des rapports de test d'un groupe de rapports](#)
- [Afficher les rapports de test dans votre AWS compte](#)

Affichage des rapports de test d'une génération

Pour afficher les rapports de test d'une génération

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Localisez la génération que vous souhaitez afficher. Si vous connaissez le projet qui a exécuté la génération qui a créé le rapport de test :
 1. Dans le panneau de navigation, choisissez Créer des projets, puis choisissez le projet avec la génération qui a exécuté le rapport de test que vous souhaitez afficher.

2. Choisissez Générer l'historique, puis choisissez la génération exécutée qui a créé les rapports que vous souhaitez afficher.

Vous pouvez également localiser la génération dans l'historique de génération de votre compte AWS :

1. Dans le panneau de navigation, choisissez Historique de génération, puis choisissez la génération qui a créé les rapports de test que vous souhaitez afficher.
3. Dans la page de génération, choisissez Rapports, puis choisissez un rapport de test pour afficher ses détails.

Affichage des rapports de test d'un groupe de rapports

Pour afficher des rapports de test dans un groupe de rapports

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le groupe de rapports qui contient les rapports de test à afficher.
4. Choisissez un rapport de test pour voir ses détails.

Afficher les rapports de test dans votre AWS compte

Pour consulter les rapports de test dans votre AWS compte

1. Ouvrez la AWS CodeBuild console sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Historique des événements.
3. Choisissez un rapport de test pour voir ses détails.

Autorisations relatives aux rapports de test

Cette rubrique décrit des informations importantes sur les autorisations liées aux rapports de test.

Rubriques

- [Rôle IAM pour les rapports de test](#)
- [Autorisations pour les opérations de rapports de test](#)
- [Exemples d'autorisations de rapport de test](#)

Rôle IAM pour les rapports de test

Pour exécuter un rapport de test et mettre à jour un projet de façon à inclure des rapports de test, votre rôle IAM requiert les autorisations suivantes. Ces autorisations sont incluses dans les politiques AWS gérées prédéfinies. Si vous souhaitez ajouter des rapports de test à un projet de génération existant, vous devez ajouter ces autorisations vous-même.

- `CreateReportGroup`
- `CreateReport`
- `UpdateReport`
- `BatchPutTestCases`

Pour exécuter un rapport de couverture de code, votre rôle IAM doit également inclure l'autorisation `BatchPutCodeCoverages`.

Note

`BatchPutTestCases`, `CreateReportUpdateReport`, et `BatchPutCodeCoverages` ne sont pas des autorisations publiques. Vous ne pouvez pas appeler une AWS CLI commande ou une méthode du SDK correspondante pour ces autorisations.

Pour vous assurer que vous disposez de ces autorisations, vous pouvez associer la politique suivante à votre rôle IAM :

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",

```

```
        "codebuild:UpdateReport",
        "codebuild:BatchPutTestCases",
        "codebuild:BatchPutCodeCoverages"
    ]
}
```

Nous vous recommandons de limiter cette stratégie aux groupes de rapports que vous devez utiliser. Ce qui suit restreint les autorisations aux seuls groupes de rapports dont les deux sont inclus ARNs dans la politique :

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Ce qui suit restreint les autorisations aux groupes de rapports créés en exécutant des versions d'un projet nommé `my-project` :

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

```
}
```

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Autorisations pour les opérations de rapports de test

Vous pouvez spécifier des autorisations pour les opérations d' API CodeBuild de rapports de test suivantes :

- BatchGetReportGroups
- BatchGetReports
- CreateReportGroup
- DeleteReportGroup
- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

Pour de plus amples informations, veuillez consulter [AWS CodeBuild référence aux autorisations](#).

Exemples d'autorisations de rapport de test

Pour de plus amples informations sur les exemples de stratégies liées aux rapports de test, veuillez consulter les rubriques suivantes :

- [Autorisation d'un utilisateur à modifier un groupe de rapports](#)
- [Autorisation d'un utilisateur à créer un groupe de rapports](#)
- [Autorisation d'un utilisateur à supprimer un rapport](#)
- [Autorisation d'un utilisateur à supprimer un groupe de rapports](#)

- [Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport](#)

Statuts des rapports de test

L'état d'un rapport de test peut être l'un des suivants :

- **GENERATING** : l'exécution des cas de test est toujours en cours.
- **DELETING** : le rapport de test est en cours de suppression. Lorsqu'un rapport de test est supprimé, ses cas de test sont également supprimés. Les fichiers de données de résultats de test bruts exportés vers un compartiment S3 ne sont pas supprimés.
- **INCOMPLETE** : le rapport de test n'a pas été réalisé. Cet état peut être renvoyé pour l'une des raisons suivantes :
 - Problème de configuration du groupe de rapports qui spécifie les cas de test de ce rapport. Par exemple, le chemin d'accès aux cas de test sous le groupe de rapports dans le fichier buildspec peut être incorrect.
 - L'utilisateur IAM qui a exécuté la génération n'a pas les autorisations nécessaires pour exécuter des tests. Pour de plus amples informations, veuillez consulter [Autorisations relatives aux rapports de test](#).
 - La génération n'a pas été terminée en raison d'une erreur qui n'est pas liée aux tests.
- **SUCCEEDED** : tous les cas de test ont été réussis.
- **FAILED** : certains des cas de test n'ont pas réussi.

Chaque cas de test renvoie un état. L'état d'un cas de test peut être l'un des suivants :

- **SUCCEEDED** : le cas de test a réussi.
- **FAILED** : le cas de test a échoué.
- **ERROR** : le cas de test a entraîné une erreur inattendue.
- **SKIPPED** : le cas de test ne s'est pas exécuté.

- UNKNOWN : le cas de test a renvoyé un état autre que SUCCEEDED, FAILED, ERROR ou SKIPPED.

Un rapport de test peut comporter un maximum de 500 résultats de cas de test. Si plus de 500 scénarios de test sont exécutés, CodeBuild hiérarchise les tests avec le statut FAILED et tronque les résultats du scénario de test.

Utilisation AWS CodeBuild avec Amazon Virtual Private Cloud

En règle générale, AWS CodeBuild impossible d'accéder aux ressources d'un VPC. Pour activer l'accès, vous devez fournir des informations de configuration supplémentaires spécifiques au VPC dans la configuration de votre CodeBuild projet. Cela inclut l'ID VPC, le sous-réseau VPC IDs et le groupe de sécurité VPC. IDs Les builds compatibles VPC peuvent ensuite accéder aux ressources à l'intérieur de votre VPC. Pour plus d'informations sur la configuration d'un VPC dans Amazon VPC, consultez le guide de l'utilisateur Amazon [VPC](#).

Rubriques

- [Cas d'utilisation](#)
- [Les meilleures pratiques pour VPCs](#)
- [Limites de VPCs](#)
- [Autorisez l'accès à Amazon VPC dans vos projets CodeBuild](#)
- [Résoudre les problèmes de configuration de votre VPC](#)
- [Utilisation de points de terminaison de VPC](#)
- [Utilisation AWS CodeBuild avec un serveur proxy géré](#)
- [Utilisation AWS CodeBuild avec un serveur proxy](#)
- [CloudFormation Modèle VPC](#)

Cas d'utilisation

La connectivité VPC depuis les AWS CodeBuild builds permet de :

- Exécutez des tests d'intégration depuis votre build par rapport aux données d'une base de données Amazon RDS isolée sur un sous-réseau privé.
- Interrogez les données d'un ElastiCache cluster Amazon directement à partir de tests.
- Interagissez avec des services Web internes hébergés sur Amazon EC2, Amazon ECS ou des services qui utilisent Elastic Load Balancing interne.
- Récupérez des dépendances de référentiels d'artefact internes auto-hébergés, comme PyPI pour Python, Maven pour Java et npm pour Node.js.

- Accédez aux objets d'un compartiment S3 configuré pour autoriser l'accès uniquement via un point de terminaison Amazon VPC.
- Recherchez des services web externes qui nécessitent des adresses IP fixes par le biais de l'adresse IP Elastic de la passerelle NAT ou d'une instance NAT associée à votre ou vos sous-réseaux.

Vos générations peuvent accéder à n'importe quelle ressource qui est hébergée dans votre VPC.

Les meilleures pratiques pour VPCs

Utilisez cette liste de contrôle lorsque vous configurez un VPC avec lequel vous souhaitez travailler. CodeBuild

- Configurez votre VPC avec des sous-réseaux publics et privés, ainsi qu'une passerelle NAT. La passerelle NAT doit résider dans un sous-réseau public. Pour plus d'informations, consultez [VPC avec des sous-réseaux publics et privés \(NAT\)](#) dans le Amazon VPC Guide de l'utilisateur.

Important

Vous avez besoin d'une passerelle NAT ou d'une instance NAT à utiliser CodeBuild avec votre VPC afin de CodeBuild pouvoir atteindre les points de terminaison publics (par exemple, pour exécuter des commandes CLI lors de l'exécution de builds). Vous ne pouvez pas utiliser la passerelle Internet à la place d'une passerelle NAT ou d'une instance NAT car elle CodeBuild ne prend pas en charge l'attribution d'adresses IP élastiques aux interfaces réseau qu'elle crée, et l'attribution automatique d'une adresse IP publique n'est pas prise en charge par Amazon EC2 pour les interfaces réseau créées en dehors des lancements d'instances Amazon EC2.

- Incluez plusieurs zones de disponibilité avec votre VPC.
- Assurez-vous qu'aucun trafic entrant (entrant) n'est autorisé à accéder à vos builds pour vos groupes de sécurité. CodeBuild n'a pas d'exigences spécifiques pour le trafic sortant, mais vous devez autoriser l'accès à toutes les ressources Internet requises pour votre build, comme GitHub Amazon S3.

Pour de plus amples informations, veuillez consulter [Règles des groupes de sécurité](#) dans le Guide de l'utilisateur Amazon VPC.

- Configurez des sous-réseaux distincts pour vos générations.

- Lorsque vous configurez vos CodeBuild projets pour accéder à votre VPC, choisissez des sous-réseaux privés uniquement.

Pour plus d'informations sur la configuration d'un VPC dans Amazon VPC, consultez le guide de l'utilisateur Amazon [VPC](#).

Pour plus d'informations sur l'utilisation CloudFormation de la fonctionnalité VPC pour configurer un CodeBuild VPC, consultez le. [CloudFormation Modèle VPC](#)

Limites de VPCs

- La connectivité VPC depuis n' CodeBuild est pas prise en charge pour le partage. VPCs

Autorisez l'accès à Amazon VPC dans vos projets CodeBuild

Incluez ces paramètres dans la configuration de votre VPC :

- Pour l'ID VPC, choisissez l'ID VPC qui utilise. CodeBuild
- Pour les sous-réseaux, choisissez un sous-réseau privé avec traduction NAT qui inclut ou possède des itinéraires vers les ressources utilisées par. CodeBuild
- Pour les groupes de sécurité, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources du VPCs.

Pour utiliser la console afin de créer un projet de génération, consultez [Création d'un projet de génération \(console\)](#). Lorsque vous créez ou modifiez votre CodeBuild projet, dans VPC, choisissez votre ID VPC, vos sous-réseaux et vos groupes de sécurité.

Pour utiliser le AWS CLI pour créer un projet de construction, voir [Création d'un projet de génération \(AWS CLI\)](#). Si vous utilisez le AWS CLI with CodeBuild, le rôle de service utilisé CodeBuild pour interagir avec les services au nom de l'utilisateur IAM doit être associé à une politique. Pour plus d'informations, consultez [Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC](#).

L'*vpcConfig* objet doit inclure votre *vpcIdsecurityGroupIds*, et *subnets*.

- **vpcId** : obligatoire. L'ID VPC que CodeBuild utilise. Exécutez cette commande pour obtenir une liste de tous les Amazon VPC de votre région :

```
aws ec2 describe-vpcs
```

- **subnets** : obligatoire. Le sous-réseau IDs qui inclut les ressources utilisées par CodeBuild. Exécutez cette commande pour obtenir les éléments suivants IDs :

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Remplacez us-east-1 par votre région.

- **securityGroupIds** : obligatoire. Le groupe de sécurité IDs utilisé par CodeBuild pour autoriser l'accès aux ressources du VPCs. Exécutez cette commande pour les obtenir IDs :

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Remplacez us-east-1 par votre région.

Résoudre les problèmes de configuration de votre VPC

Utilisez les informations affichées dans le message d'erreur pour vous aider à identifier, à diagnostiquer et à résoudre les problèmes.

Voici quelques instructions pour vous aider lors du dépannage d'une erreur VPC CodeBuild courante : Build does not have internet connectivity. Please check subnet network configuration.

1. [Assurez-vous que votre passerelle Internet est attachée au VPC.](#)
2. [Assurez-vous que la table de routage de votre sous-réseau public pointe vers la passerelle Internet.](#)
3. [Assurez-vous que votre réseau ACLs autorise la circulation du trafic.](#)

4. [Assurez-vous que vos groupes de sécurité autorisent la circulation du trafic.](#)
5. [Dépannez votre passerelle NAT.](#)
6. [Assurez-vous que la table de routage des sous-réseaux privés pointe vers la passerelle NAT.](#)
7. Assurez-vous que le rôle de service utilisé CodeBuild pour interagir avec les services au nom de l'utilisateur IAM dispose des autorisations définies dans [cette politique](#). Pour de plus amples informations, veuillez consulter [CodeBuild Autoriser l'interaction avec d'autres AWS services](#).

S'il CodeBuild il manque des autorisations, il se peut que vous receviez un message d'erreur indiquant, `Unexpected EC2 error: UnauthorizedOperation`. Cette erreur peut se produire s'il CodeBuild ne dispose pas des autorisations Amazon EC2 requises pour travailler avec un VPC.

Utilisation de points de terminaison de VPC

Vous pouvez améliorer la sécurité de vos builds en les configurant de manière AWS CodeBuild à utiliser un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par PrivateLink une technologie que vous pouvez utiliser pour accéder en privé à Amazon EC2 CodeBuild et en utilisant des adresses IP privées. PrivateLink restreint tout le trafic réseau entre vos instances gérées et Amazon EC2 vers le réseau Amazon. CodeBuild (Les instances gérées n'ont pas accès à Internet). De même, vous n'avez pas besoin d'une passerelle Internet, d'un périphérique NAT ni d'une passerelle privée virtuelle. Il n'est pas obligatoire de configurer PrivateLink, mais c'est recommandé. [Pour plus d'informations sur les points PrivateLink de terminaison VPC, consultez Qu'est-ce que c'est ? AWS PrivateLink](#).

Avant de créer des points de terminaison de VPC

Avant de configurer les points de terminaison VPC pour AWS CodeBuild, tenez compte des restrictions et limitations suivantes.

Note

Utilisez une [passerelle NAT](#) si vous souhaitez utiliser CodeBuild des AWS services qui ne prennent pas en charge les connexions Amazon VPC PrivateLink.

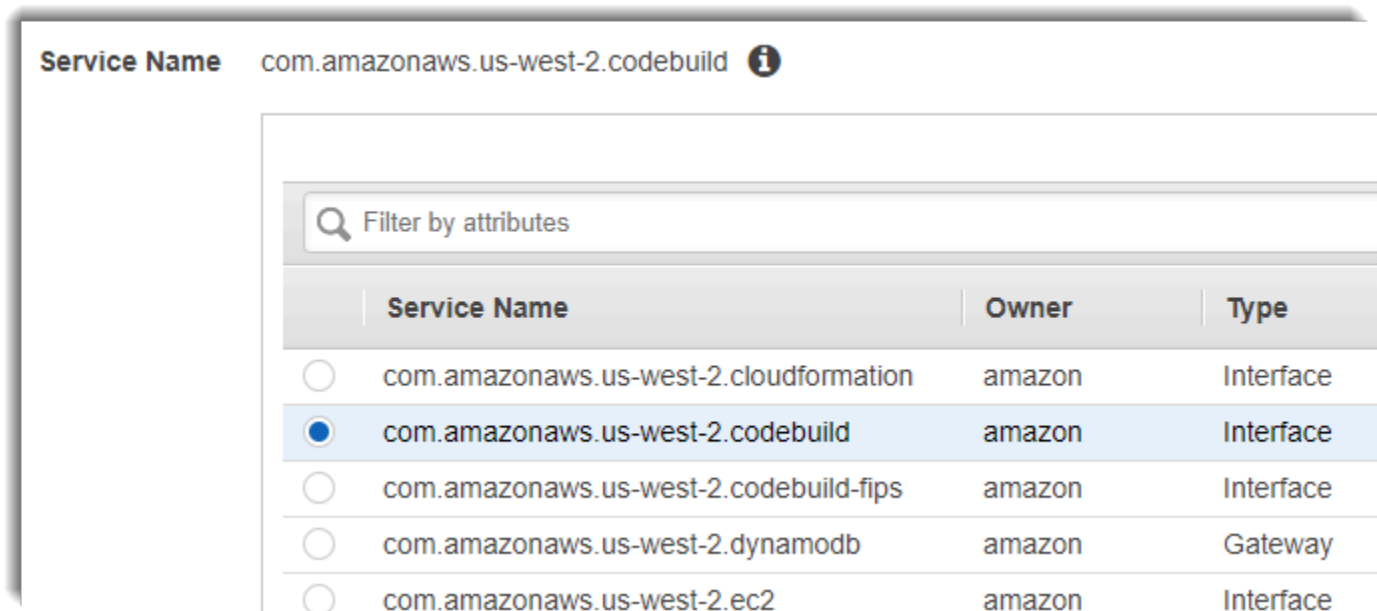
- Les points de terminaison VPC prennent en charge le DNS fourni par Amazon via Amazon Route 53 uniquement. Si vous souhaitez utiliser votre propre DNS, vous pouvez utiliser le transfert DNS

conditionnel. Pour plus d'informations, consultez [Jeux d'options DHCP](#) dans le Guide de l'utilisateur Amazon VPC.

- Les points de terminaison de VPC ne prennent pas en charge les demandes inter-régionales pour le moment. Assurez-vous de créer votre point de terminaison dans la même AWS région que tous les compartiments S3 qui stockent les entrées et sorties de votre build. Vous pouvez utiliser la console Amazon S3 ou la [get-bucket-location](#) commande pour trouver l'emplacement de votre compartiment. Utilisez un point de terminaison Amazon S3 spécifique à une région pour accéder à votre compartiment (par exemple, `<bucket-name>.s3-us-west-2.amazonaws.com`). Pour plus d'informations sur les points de terminaison spécifiques à une région pour Amazon S3, consultez [Amazon Simple Storage Service](#) dans le. Référence générale d'Amazon Web Services Si vous utilisez le AWS CLI pour envoyer des demandes à Amazon S3, définissez votre région par défaut sur la même région que celle dans laquelle votre compartiment a été créé, ou utilisez le `--region` paramètre dans vos demandes.

Créez des points de terminaison VPC pour CodeBuild

Suivez les instructions indiquées dans [Création d'un point de terminaison d'interface](#) pour créer le point de terminaison `com.amazonaws.us-west-2.codebuild`. Il s'agit d'un point de terminaison VPC pour AWS CodeBuild



region représente l'identifiant de région d'une AWS région prise en charge par CodeBuild, par exemple `us-east-2` pour la région USA Est (Ohio). Pour obtenir la liste des AWS régions prises en charge, reportez-vous [CodeBuild](#) à la section Référence AWS générale. Le point de terminaison

est prérempli avec la région que vous avez spécifiée lorsque vous vous êtes connecté AWS. Si vous changez de région, le point de terminaison du VPC est mis à jour en conséquence.

Créez une politique de point de terminaison VPC pour CodeBuild

Vous pouvez créer une politique pour les points de terminaison Amazon VPC AWS CodeBuild dans laquelle vous pouvez spécifier :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources qui peuvent avoir des actions exécutées sur elles.

L'exemple de stratégie suivant spécifie que tous les mandataires peuvent uniquement lancer et afficher des générations pour le projet `project-name`.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Pour plus d'informations, consultez [Contrôle de l'accès aux services avec points de terminaison d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Utilisation AWS CodeBuild avec un serveur proxy géré

Pour exécuter des flottes de capacité AWS CodeBuild réservée sur un serveur proxy géré, vous devez configurer le serveur proxy pour autoriser ou refuser le trafic à destination et en provenance de sites externes à l'aide de règles de proxy. Notez que l'exécution de flottes de capacité réservée sur un serveur proxy géré n'est pas prise en charge pour VPC, Windows ou macOS.

⚠ Important

Il existe des coûts supplémentaires basés sur la durée pendant laquelle une configuration proxy est présente dans la flotte. Pour plus d'informations, consultez la section <https://aws.amazon.com/codebuild/tarification/>.

Rubriques

- [Configurer une configuration de proxy géré pour les flottes de capacité réservée](#)
- [Gérez un parc à capacité CodeBuild réservée](#)

Configurer une configuration de proxy géré pour les flottes de capacité réservée

Pour configurer un serveur proxy géré pour votre parc de capacités réservées, vous devez activer cette fonctionnalité lors de la création de votre parc dans votre console ou à l'aide du AWS CLI. Vous devez définir plusieurs propriétés :

Définir les configurations de proxy (facultatif)

Configurations de proxy qui appliquent un contrôle d'accès réseau à vos instances de capacité réservée.

Comportement par défaut

Définit le comportement du trafic sortant.

Allow

Autorise le trafic sortant vers toutes les destinations par défaut.

Refuser

Refuse le trafic sortant vers toutes les destinations par défaut.

Règles relatives au proxy

Spécifie les domaines de destination auxquels le contrôle d'accès réseau doit être restreint.

Pour définir les configurations de proxy dans votre console, consultez [Création d'un parc de capacités réservées](#) les instructions. Pour définir des configurations de proxy à l'aide du AWS CLI, vous pouvez le faire en modifiant la syntaxe JSON suivante et en enregistrant vos résultats :

```
"proxyConfiguration": {
  "defaultBehavior": "ALLOW_ALL" | "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN" | "IP",
      "effect": "ALLOW" | "DENY",
      "entities": [
        "destination"
      ]
    }
  ]
}
```

Votre fichier JSON peut ressembler à ce qui suit :

```
"proxyConfiguration": {
  "defaultBehavior": "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN",
      "effect": "ALLOW",
      "entities": [
        "github.com"
      ]
    }
  ]
}
```

Gérez un parc à capacité CodeBuild réservée

Lorsque vous exécutez des flottes de capacité AWS CodeBuild réservée avec votre serveur proxy géré, il CodeBuild définira automatiquement ses variables HTTP_PROXY et celles de HTTPS_PROXY l'environnement avec les adresses proxy gérées. Si votre logiciel de dépendance possède sa propre configuration et ne respecte pas les variables d'environnement, vous pouvez vous référer à ces valeurs et mettre à jour votre configuration logicielle dans vos commandes de génération afin d'acheminer correctement votre trafic de génération via le proxy géré. Pour plus d'informations,

consultez [Créez un projet de construction dans AWS CodeBuild](#) et [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).

Utilisation AWS CodeBuild avec un serveur proxy

Vous pouvez l'utiliser AWS CodeBuild avec un serveur proxy pour réguler le trafic HTTP et HTTPS à destination et en provenance d'Internet. Pour fonctionner CodeBuild avec un serveur proxy, vous devez installer un serveur proxy dans un sous-réseau public et CodeBuild dans un sous-réseau privé d'un VPC.

Il existe deux principaux cas d'utilisation pour l'exécution CodeBuild sur un serveur proxy :

- Il élimine l'utilisation d'une passerelle NAT ou d'une instance NAT dans votre VPC.
- Il vous permet de spécifier les accès URLs auxquels les instances du serveur proxy peuvent accéder et URLs celles auxquelles le serveur proxy refuse l'accès.

Vous pouvez utiliser CodeBuild deux types de serveurs proxy. Dans les deux cas, le serveur proxy s'exécute dans un sous-réseau public et CodeBuild dans un sous-réseau privé.

- Proxy explicite : si vous utilisez un serveur proxy explicite, vous devez configurer `NO_PROXYHTTP_PROXY`, et les variables d'`HTTPS_PROXY` environnement CodeBuild au niveau du projet. Pour plus d'informations, consultez [Modifier les paramètres du projet de construction dans AWS CodeBuild](#) et [Créez un projet de construction dans AWS CodeBuild](#).
- Proxy transparent : si vous utilisez un serveur proxy transparent, aucune configuration spéciale n'est nécessaire.

Rubriques

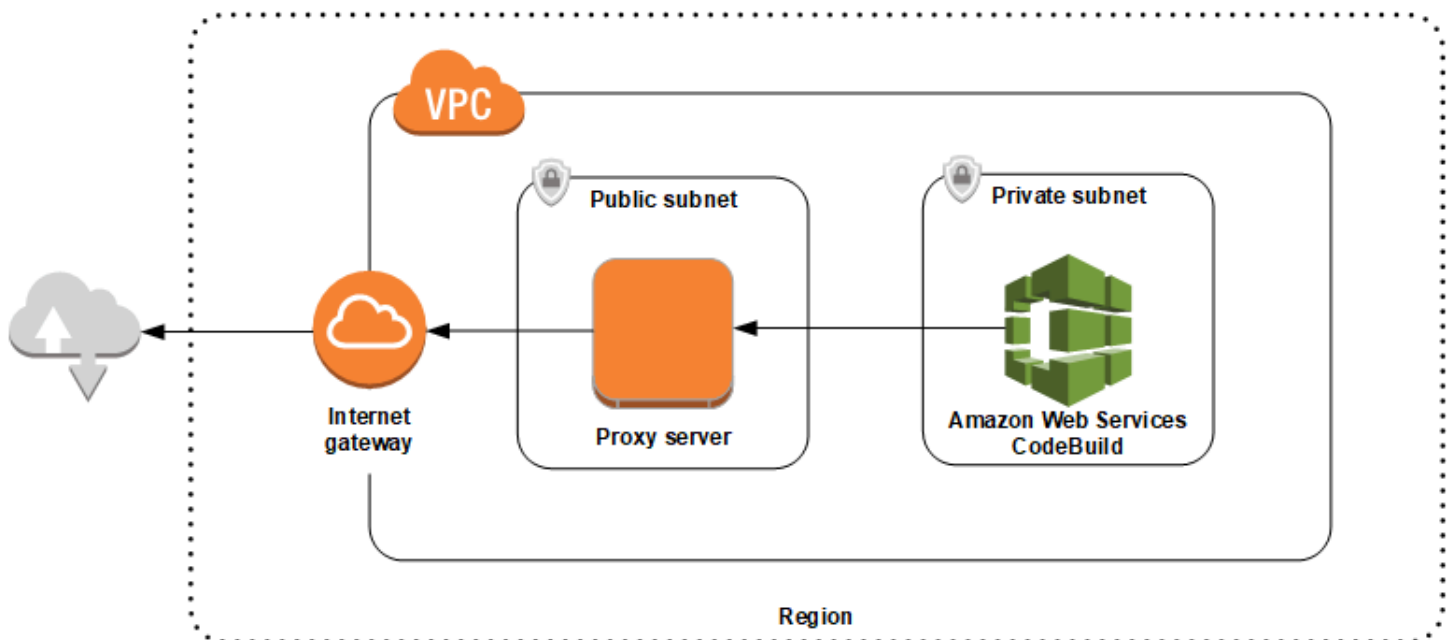
- [Configuration des composants nécessaires à l'exécution CodeBuild sur un serveur proxy](#)
- [Exécuter CodeBuild sur un serveur proxy explicite](#)
- [Exécuter CodeBuild sur un serveur proxy transparent](#)
- [Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy](#)

Configuration des composants nécessaires à l'exécution CodeBuild sur un serveur proxy

Les composants suivants sont nécessaires pour fonctionner AWS CodeBuild sur un serveur proxy transparent ou explicite :

- Un VPC.
- Un sous-réseau public dans votre VPC pour le serveur proxy
- Un sous-réseau privé dans votre VPC pour CodeBuild
- Une passerelle Internet qui permet la communication entre le VPC et Internet.

Le schéma suivant illustre la manière dont les deux composants interagissent.



Configuration d'un VPC, de sous-réseaux et d'une passerelle réseau

Les étapes suivantes sont requises pour fonctionner AWS CodeBuild sur un serveur proxy transparent ou explicite.

1. Créez un VPC. Pour plus d'informations, consultez [Création d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.
2. Créez deux sous-réseaux dans votre VPC. L'un des deux est un sous-réseau public, nommé `Public Subnet`, dans lequel le serveur proxy s'exécute. L'autre est un sous-réseau privé nommé `Private Subnet` dans lequel CodeBuild s'exécute.

Pour de plus amples informations, veuillez consulter [Création d'un sous-réseau dans votre VPC](#).

3. Créez et attachez une passerelle Internet à votre VPC. Pour de plus amples informations, veuillez consulter [Création et attachement d'une passerelle Internet](#).
4. Ajoutez une règle à la table de routage par défaut qui achemine le trafic sortant à partir du VPC (0.0.0.0/0) vers la passerelle Internet. Pour de plus amples informations, veuillez consulter [Ajout et retrait de routes d'une table de routage](#).
5. Ajoutez une règle au groupe de sécurité par défaut de votre VPC qui autorise le trafic SSH entrant (TCP 22) à partir de votre VPC (0.0.0.0/0).
6. Suivez les instructions de la section [Lancement d'une instance à l'aide de l'assistant de lancement d'instance](#) du guide de l'utilisateur Amazon EC2 pour lancer une instance Amazon Linux. Lorsque vous exécutez l'assistant, choisissez les options suivantes :
 - Dans Choisir un type d'instance, choisissez une Amazon Linux Amazon Machine Image (AMI).
 - Dans Subnet (Sous-réseau), choisissez le sous-réseau public que vous avez créé précédemment dans cette rubrique. Si vous avez utilisé le nom suggéré, il s'agit de Public Subnet.
 - Pour Auto-assign Public IP (Attribuer automatiquement) l'adresse IP publique, sélectionnez (Enable) Activer.
 - Sur la page Configure Security Group (Configurer un groupe de sécurité), pour Assign a security group (Affecter un groupe de sécurité), choisissez Select an existing security group (Sélectionner un groupe de sécurité existant). Ensuite, choisissez le groupe de sécurité par défaut.
 - Après avoir choisi Launch (Lancer), choisissez une paire de clés existante ou créez-en une.

Choisissez les paramètres par défaut pour toutes les autres options.

7. Une fois que votre instance EC2 est en cours d'exécution, désactivez les source/destination vérifications. Pour plus d'informations, consultez la section [Désactivation Source/Destination des contrôles](#) dans le guide de l'utilisateur Amazon VPC.
8. Créez une table de routage dans votre VPC. Ajoutez une règle à la table de routage qui achemine le trafic destiné à Internet vers votre serveur proxy. Associez cette table de routage à votre sous-réseau privé. Cela est nécessaire pour que les demandes sortantes provenant des instances de votre sous-réseau privé, où CodeBuild s'exécute, soient toujours acheminées via le serveur proxy.

Installation et configuration d'un serveur proxy

Il existe de nombreux serveurs proxy parmi lesquels choisir. Un serveur proxy open source, Squid, est utilisé ici pour montrer comment AWS CodeBuild fonctionne un serveur proxy. Vous pouvez appliquer les mêmes concepts à d'autres serveurs proxy.

Pour installer Squid, utilisez un référentiel yum en exécutant les commandes suivantes :

```
sudo yum update -y
sudo yum install -y squid
```

Après avoir installé Squid, modifiez son fichier `squid.conf` en suivant les instructions plus loin dans cette rubrique.

Configuration de Squid pour le trafic HTTPS

Pour HTTPS, le trafic HTTP est encapsulé dans une connexion à protocole TLS (Transport Layer Security, Sécurité de la couche transport). Squid utilise une fonctionnalité appelée [SslPeekAndSplice](#) pour récupérer l'indication du nom du serveur (SNI) à partir de l'initiation TLS qui contient l'hôte Internet demandé. Cette opération est requise pour que Squid ne soit pas obligé de déchiffrer le trafic HTTPS. Pour l'activer `SslPeekAndSplice`, Squid a besoin d'un certificat. Créez ce certificat à l'aide d'OpenSSL :

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

Pour HTTP, Squid ne nécessite aucune configuration. À partir de tous les messages de demande HTTP/1.1, il peut extraire le champ d'en-tête de l'hôte, qui spécifie l'hôte Internet demandé.

Exécuter CodeBuild sur un serveur proxy explicite

Pour fonctionner AWS CodeBuild sur un serveur proxy explicite, vous devez configurer le serveur proxy pour autoriser ou refuser le trafic vers et depuis des sites externes, puis configurer les variables d'HTTPS_PROXY et HTTP_PROXY.

Rubriques

- [Configuration de Squid en tant que serveur proxy explicite](#)
- [Création d'un CodeBuild projet](#)
- [Exemple de fichier squid.conf de serveur proxy explicite](#)

Configuration de Squid en tant que serveur proxy explicite

Pour configurer le serveur proxy Squid en tant que serveur explicite, vous devez apporter les modifications suivantes au fichier `/etc/squid/squid.conf` :

- Supprimez les règles de liste de contrôle d'accès (ACL) par défaut suivantes :

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Ajoutez les règles suivantes à la place des règles ACL par défaut que vous avez supprimées. La première ligne autorise les demandes à partir de votre VPC. Les deux lignes suivantes permettent à votre serveur proxy d'accéder à une destination URLs susceptible d'être utilisée par AWS CodeBuild. Modifiez l'expression régulière dans la dernière ligne pour spécifier des compartiments S3 ou un CodeCommit référentiel dans une AWS région. Par exemple :

- Si votre source est Amazon S3, utilisez la commande `acl download_src dstdomain_regex .*s3\.us-west-1\.amazonaws\.com` pour autoriser l'accès aux compartiments S3 de la `us-west-1` région.
- Si votre source l'est AWS CodeCommit, `git-codecommit.<your-region>.amazonaws.com` utilisez-le pour ajouter une AWS région à une liste autorisée.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
```



```
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from
Amazon S3 or CodeCommit
```

- Remplacez `http_access allow localnet` par ce qui suit :

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Si vous souhaitez que votre build charge des journaux et des artefacts, effectuez l'une des actions suivantes :

1. Avant l'instruction `http_access deny all`, insérez les instructions suivantes. Ils permettent CodeBuild d'accéder CloudWatch à Amazon S3. L'accès à CloudWatch est requis pour CodeBuild créer des CloudWatch journaux. L'accès à Amazon S3 est requis pour le téléchargement d'artefacts et la mise en cache d'Amazon S3.

- ```
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

- Après avoir enregistré `squid.conf`, exécutez la commande suivante :

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Ajoutez proxy à votre fichier `buildspec`. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

```
version: 0.2
proxy:
 upload-artifacts: yes
 logs: yes
phases:
 build:
 commands:
```

- command

### Note

Si vous recevez un message d'erreur `RequestError` de temporisation, consultez [RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy](#).

Pour plus d'informations, consultez [Exemple de fichier `squid.conf` de serveur proxy explicite](#) plus loin dans cette rubrique.

## Création d'un CodeBuild projet

Pour fonctionner AWS CodeBuild avec votre serveur proxy explicite, définissez ses variables `HTTP_PROXY` et ses variables d'`HTTPS_PROXY` environnement avec l'adresse IP privée de l'instance EC2 que vous avez créée pour votre serveur proxy et le port 3128 au niveau du projet. L'adresse IP privée se présente ainsi : `http://your-ec2-private-ip-address:3128`. Pour plus d'informations, consultez [Créez un projet de construction dans AWS CodeBuild](#) et [Modifier les paramètres du projet de construction dans AWS CodeBuild](#).

Utilisez la commande suivante pour afficher le journal d'accès proxy Squid :

```
sudo tail -f /var/log/squid/access.log
```

## Exemple de fichier **squid.conf** de serveur proxy explicite

L'exemple suivant présente un fichier `squid.conf` configuré pour un serveur proxy explicite.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
```

```
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
#
Recommended minimum Access Permission configuration:
#
Deny requests to certain unsafe ports
http_access deny !Safe_ports
Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
We strongly recommend the following be uncommented to protect innocent
web applications running on the proxy server who think the only
one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
Example rule allowing access from your local networks.
Adapt localnet in the ACL section to list your (internal) IP networks
from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
```

```
ssl_bump terminate step2 all
And finally deny all other access to this proxy
http_access deny all
Squid normally listens to port 3128
http_port 3128
Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

## Exécuter CodeBuild sur un serveur proxy transparent

Pour fonctionner AWS CodeBuild sur un serveur proxy transparent, vous devez configurer le serveur proxy de manière à ce qu'il puisse accéder aux sites Web et aux domaines avec lesquels il interagit.

### Rubriques

- [Configuration de Squid en tant que serveur proxy transparent](#)
- [Création d'un CodeBuild projet](#)

## Configuration de Squid en tant que serveur proxy transparent

Pour configurer un serveur proxy en tant que serveur transparent, vous devez lui accorder l'accès aux domaines et aux sites Web auxquels vous souhaitez qu'il accède. Pour fonctionner AWS CodeBuild avec un serveur proxy transparent, vous devez lui accorder l'accès à `amazonaws.com`. Vous devez également autoriser l'accès à d'autres CodeBuild utilisations du site Web. Elles varient en fonction de la façon dont vous créez vos CodeBuild projets. Les sites Web relatifs à des référentiels tels que Bitbucket GitHub, Yum et Maven sont des exemples de sites Web. Pour accorder à Squid l'accès à des domaines et à des sites Web spécifiques, utilisez une commande similaire à la suivante pour mettre à jour le fichier `squid.conf`. Cet exemple de commande accorde l'accès à `amazonaws.com`, `github.com` et `bitbucket.com`. Vous pouvez modifier cet exemple pour donner accès à d'autres sites Web.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
EOF
```

Les demandes entrantes à partir des instances du sous-réseau privé doivent être redirigées vers les ports Squid. Squid écoute le trafic HTTP sur le port 3129 (au lieu de 80) et le trafic HTTPS sur le port 3130 (au lieu de 443). Utilisez la commande iptables pour acheminer le trafic :

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

## Création d'un CodeBuild projet

Après avoir configuré votre serveur proxy, vous pouvez l'utiliser AWS CodeBuild dans un sous-réseau privé sans autre configuration. Chaque demande HTTP et HTTPS passe par le serveur proxy public. Utilisez la commande suivante pour afficher le journal d'accès proxy Squid :

```
sudo tail -f /var/log/squid/access.log
```

## Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy

Utilisez les procédures suivantes pour exécuter un gestionnaire de packages et d'autres outils sur un serveur proxy.

Pour exécuter un outil, tel qu'un gestionnaire de packages, sur un serveur proxy

1. Pour inclure l'outil dans la liste d'autorisation de votre serveur proxy, ajoutez des instructions dans le fichier `squid.conf`.
2. Ajoutez une ligne à votre fichier `buildspec` qui pointe vers le point de terminaison privé de votre serveur proxy.

Les exemples suivants montrent comment procéder pour `apt-get`, `curl` et `maven`. Si vous utilisez un outil différent, les mêmes principes s'appliquent. Ajoutez-le à une liste d'autorisation dans le `squid.conf` fichier et ajoutez une commande à votre fichier `buildspec` pour identifier le point de CodeBuild terminaison de votre serveur proxy.

### Exécution de **apt-get** sur un serveur proxy

1. Ajoutez les instructions suivantes à votre fichier `squid.conf` pour ajouter `apt-get` à une liste d'autorisation sur votre serveur proxy. Les trois premières lignes permettent `apt-get` de fonctionner dans l'environnement de construction.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Ajoutez l'instruction suivante dans votre fichier `buildspec` afin que les commandes `apt-get` recherchent la configuration proxy dans `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

## Exécution de **curl** sur un serveur proxy

1. Insérez ce qui suit dans votre fichier `squid.conf` pour ajouter `curl` à une liste d'autorisation dans votre environnement de génération.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a webiste. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Ajoutez l'instruction suivante dans votre fichier `buildspec` de sorte que `curl` utilise le serveur proxy privé pour accéder au site Web que vous avez ajouté à `squid.conf`. Dans cet exemple, le site Web est `google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

## Exécution de **maven** sur un serveur proxy

1. Insérez ce qui suit dans votre fichier `squid.conf` pour ajouter `maven` à une liste d'autorisation dans votre environnement de génération.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Ajoutez l'instruction suivante à votre fichier `buildspec`.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

## CloudFormation Modèle VPC

CloudFormation vous permet de créer et de provisionner des déploiements d' AWS infrastructure de manière prévisible et répétée, en utilisant des fichiers modèles pour créer et supprimer un ensemble de ressources en une seule unité (une pile). Pour plus d'informations, consultez le [CloudFormation Guide de l'utilisateur](#).

Ce qui suit est un modèle CloudFormation YAML pour configurer un VPC à utiliser. AWS CodeBuild Ce fichier est également disponible dans le [fichier samples.zip](#).

Description: This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

### Parameters:

#### EnvironmentName:

Description: An environment name that is prefixed to resource names  
Type: String

#### VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC  
Type: String  
Default: 10.192.0.0/16

#### PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone  
Type: String  
Default: 10.192.10.0/24

#### PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone  
Type: String  
Default: 10.192.11.0/24



**PrivateSubnet1CIDR:**

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 10.192.20.0/24

**PrivateSubnet2CIDR:**

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 10.192.21.0/24

**Resources:****VPC:**

Type: AWS::EC2::VPC

**Properties:**

CidrBlock: !Ref VpcCIDR

EnableDnsSupport: true

EnableDnsHostnames: true

**Tags:**

- Key: Name

Value: !Ref EnvironmentName

**InternetGateway:**

Type: AWS::EC2::InternetGateway

**Properties:****Tags:**

- Key: Name

Value: !Ref EnvironmentName

**InternetGatewayAttachment:**

Type: AWS::EC2::VPCGatewayAttachment

**Properties:**

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref VPC

**PublicSubnet1:**

Type: AWS::EC2::Subnet

**Properties:**

VpcId: !Ref VPC

AvailabilityZone: !Select [ 0, !GetAZs '' ]

CidrBlock: !Ref PublicSubnet1CIDR

MapPublicIpOnLaunch: true

## Tags:

- Key: Name  
Value: !Sub \${EnvironmentName} Public Subnet (AZ1)

## PublicSubnet2:

Type: AWS::EC2::Subnet

## Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 1, !GetAZs '' ]

CidrBlock: !Ref PublicSubnet2CIDR

MapPublicIpOnLaunch: true

## Tags:

- Key: Name  
Value: !Sub \${EnvironmentName} Public Subnet (AZ2)

## PrivateSubnet1:

Type: AWS::EC2::Subnet

## Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 0, !GetAZs '' ]

CidrBlock: !Ref PrivateSubnet1CIDR

MapPublicIpOnLaunch: false

## Tags:

- Key: Name  
Value: !Sub \${EnvironmentName} Private Subnet (AZ1)

## PrivateSubnet2:

Type: AWS::EC2::Subnet

## Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 1, !GetAZs '' ]

CidrBlock: !Ref PrivateSubnet2CIDR

MapPublicIpOnLaunch: false

## Tags:

- Key: Name  
Value: !Sub \${EnvironmentName} Private Subnet (AZ2)

## NatGateway1EIP:

Type: AWS::EC2::EIP

DependsOn: InternetGatewayAttachment

## Properties:

Domain: vpc

## NatGateway2EIP:

```
Type: AWS::EC2::EIP
DependsOn: InternetGatewayAttachment
Properties:
 Domain: vpc
```

```
NatGateway1:
 Type: AWS::EC2::NatGateway
 Properties:
 AllocationId: !GetAtt NatGateway1EIP.AllocationId
 SubnetId: !Ref PublicSubnet1
```

```
NatGateway2:
 Type: AWS::EC2::NatGateway
 Properties:
 AllocationId: !GetAtt NatGateway2EIP.AllocationId
 SubnetId: !Ref PublicSubnet2
```

```
PublicRouteTable:
 Type: AWS::EC2::RouteTable
 Properties:
 VpcId: !Ref VPC
 Tags:
 - Key: Name
 Value: !Sub ${EnvironmentName} Public Routes
```

```
DefaultPublicRoute:
 Type: AWS::EC2::Route
 DependsOn: InternetGatewayAttachment
 Properties:
 RouteTableId: !Ref PublicRouteTable
 DestinationCidrBlock: 0.0.0.0/0
 GatewayId: !Ref InternetGateway
```

```
PublicSubnet1RouteTableAssociation:
 Type: AWS::EC2::SubnetRouteTableAssociation
 Properties:
 RouteTableId: !Ref PublicRouteTable
 SubnetId: !Ref PublicSubnet1
```

```
PublicSubnet2RouteTableAssociation:
 Type: AWS::EC2::SubnetRouteTableAssociation
 Properties:
 RouteTableId: !Ref PublicRouteTable
 SubnetId: !Ref PublicSubnet2
```

```
PrivateRouteTable1:
 Type: AWS::EC2::RouteTable
 Properties:
 VpcId: !Ref VPC
 Tags:
 - Key: Name
 Value: !Sub ${EnvironmentName} Private Routes (AZ1)

DefaultPrivateRoute1:
 Type: AWS::EC2::Route
 Properties:
 RouteTableId: !Ref PrivateRouteTable1
 DestinationCidrBlock: 0.0.0.0/0
 NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:
 Type: AWS::EC2::SubnetRouteTableAssociation
 Properties:
 RouteTableId: !Ref PrivateRouteTable1
 SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
 Type: AWS::EC2::RouteTable
 Properties:
 VpcId: !Ref VPC
 Tags:
 - Key: Name
 Value: !Sub ${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:
 Type: AWS::EC2::Route
 Properties:
 RouteTableId: !Ref PrivateRouteTable2
 DestinationCidrBlock: 0.0.0.0/0
 NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
 Type: AWS::EC2::SubnetRouteTableAssociation
 Properties:
 RouteTableId: !Ref PrivateRouteTable2
 SubnetId: !Ref PrivateSubnet2
```

```
NoIngressSecurityGroup:
 Type: AWS::EC2::SecurityGroup
 Properties:
 GroupName: "no-ingress-sg"
 GroupDescription: "Security group with no ingress rule"
 VpcId: !Ref VPC

Outputs:
 VPC:
 Description: A reference to the created VPC
 Value: !Ref VPC

 PublicSubnets:
 Description: A list of the public subnets
 Value: !Join [",", [!Ref PublicSubnet1, !Ref PublicSubnet2]]

 PrivateSubnets:
 Description: A list of the private subnets
 Value: !Join [",", [!Ref PrivateSubnet1, !Ref PrivateSubnet2]]

 PublicSubnet1:
 Description: A reference to the public subnet in the 1st Availability Zone
 Value: !Ref PublicSubnet1

 PublicSubnet2:
 Description: A reference to the public subnet in the 2nd Availability Zone
 Value: !Ref PublicSubnet2

 PrivateSubnet1:
 Description: A reference to the private subnet in the 1st Availability Zone
 Value: !Ref PrivateSubnet1

 PrivateSubnet2:
 Description: A reference to the private subnet in the 2nd Availability Zone
 Value: !Ref PrivateSubnet2

 NoIngressSecurityGroup:
 Description: Security group with no ingress rule
 Value: !Ref NoIngressSecurityGroup
```

# Connexion et surveillance AWS CodeBuild

La journalisation et la surveillance jouent un rôle important dans le maintien de la fiabilité, de la disponibilité AWS CodeBuild et des performances de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant. AWS fournit les outils suivants pour surveiller vos CodeBuild ressources et vos builds et pour répondre aux incidents potentiels.

## Rubriques

- [Enregistrez les appels AWS CodeBuild d'API avec AWS CloudTrail](#)
- [Surveillez CodeBuild les builds avec CloudWatch](#)

## Enregistrez les appels AWS CodeBuild d'API avec AWS CloudTrail

AWS CodeBuild est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans CodeBuild. CloudTrail capture tous les appels d'API CodeBuild sous forme d'événements, y compris les appels depuis la CodeBuild console et les appels de code vers le CodeBuild APIs. Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment S3, y compris les événements pour CodeBuild. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite CodeBuild, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

## Rubriques

- [À propos AWS CodeBuild des informations contenues dans CloudTrail](#)
- [À propos des entrées du fichier AWS CodeBuild journal](#)

## À propos AWS CodeBuild des informations contenues dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans CodeBuild, cette activité est enregistrée dans un CloudTrail événement avec d'autres

événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#) dans le Guide de AWS CloudTrail l'utilisateur.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour CodeBuild, créez un parcours. Un journal permet CloudTrail de fournir des fichiers journaux à un compartiment S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions . Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment S3 que vous spécifiez. Vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour en savoir plus, consultez :

- [Présentation de la création d'un journal d'activité](#)
- [CloudTrail services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Toutes les CodeBuild actions sont enregistrées CloudTrail et documentées dans la [référence de l'CodeBuild API](#). Par exemple, les appels aux actions CreateProject (dans le AWS CLI, create-project), StartBuild (dans le AWS CLI, start-project) et UpdateProject (dans le AWS CLI, update-project) génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'[élément CloudTrail UserIdentity dans le Guide](#) de l'AWS CloudTrail utilisateur.

## À propos des entrées du fichier AWS CodeBuild journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

### Note

Pour protéger les informations sensibles, les informations suivantes sont masquées dans CodeBuild les journaux :

- AWS clé d'accès IDs. Pour plus d'informations, consultez [la section Gestion des clés d'accès pour les utilisateurs IAM](#) dans le guide de l'Gestion des identités et des accès AWS utilisateur.
- Chaînes spécifiées à l'aide du stockage de paramètres. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.
- Chaînes spécifiées à l'aide de AWS Secrets Manager. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

L'exemple suivant montre une entrée de CloudTrail journal illustrant la création d'un projet de construction dans CodeBuild.

```
{
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "FederatedUser",
 "principalId": "account-ID:user-name",
 "arn": "arn:aws:sts::account-ID:federated-user/user-name",
 "accountId": "account-ID",
 "accessKeyId": "access-key-ID",
 "sessionContext": {
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2016-09-06T17:59:10Z"
 }
 }
 }
}
```



```

 },
 "sessionIssuer": {
 "type": "IAMUser",
 "principalId": "access-key-ID",
 "arn": "arn:aws:iam::account-ID:user/user-name",
 "accountId": "account-ID",
 "userName": "user-name"
 }
 },
 "eventTime": "2016-09-06T17:59:11Z",
 "eventSource": "codebuild.amazonaws.com",
 "eventName": "CreateProject",
 "awsRegion": "region-ID",
 "sourceIPAddress": "127.0.0.1",
 "userAgent": "user-agent",
 "requestParameters": {
 "awsActId": "account-ID"
 },
 "responseElements": {
 "project": {
 "environment": {
 "image": "image-ID",
 "computeType": "BUILD_GENERAL1_SMALL",
 "type": "LINUX_CONTAINER",
 "environmentVariables": []
 },
 "name": "codebuild-demo-project",
 "description": "This is my demo project",
 "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project:project-ID",
 "encryptionKey": "arn:aws:kms:region-ID:key-ID",
 "timeoutInMinutes": 10,
 "artifacts": {
 "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
 "type": "S3",
 "packaging": "ZIP",
 "outputName": "MyOutputArtifact.zip"
 },
 "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
 "lastModified": "Sep 6, 2016 10:59:11 AM",
 "source": {
 "type": "GITHUB",
 "location": "https://github.com/my-repo.git"
 }
 }
 }
}

```

```
 },
 "created": "Sep 6, 2016 10:59:11 AM"
 },
 },
 "requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
 "eventID": "581f7dd1-8d2e-40b0-aeee-0dbf7EXAMPLE",
 "eventType": "AwsApiCall",
 "recipientAccountId": "account-ID"
}
```

## Surveillez CodeBuild les builds avec CloudWatch

Vous pouvez utiliser Amazon CloudWatch pour surveiller vos builds, signaler un problème et prendre des mesures automatiques le cas échéant. Vous pouvez surveiller vos générations à deux niveaux :

### Niveau du projet

Ces mesures concernent toutes les versions du projet spécifié. Pour afficher les métriques d'un projet, spécifiez le `ProjectName` de la dimension dans CloudWatch.

### AWS niveau du compte

Ces statistiques concernent toutes les versions d'un compte. Pour voir les métriques au niveau du compte AWS , n'indiquez pas de dimension dans CloudWatch. Les indicateurs d'utilisation des ressources de construction ne sont pas disponibles au niveau du AWS compte.

CloudWatch les métriques montrent le comportement de vos builds au fil du temps. Par exemple, vous pouvez surveiller :

- Combien de versions ont été tentées dans un projet de build ou un AWS compte au fil du temps.
- Combien de versions ont été réussies dans un projet de construction ou un AWS compte au fil du temps.
- Combien de versions ont échoué dans un projet de construction ou un AWS compte au fil du temps.
- Combien de temps CodeBuild passé à exécuter des builds dans le cadre d'un projet de build ou d'un AWS compte au fil du temps.

- Optimisez l'utilisation des ressources pour une construction ou un projet de construction complet. Les indicateurs d'utilisation des ressources du build incluent des indicateurs tels que l'utilisation du processeur, de la mémoire et du stockage.

Pour de plus amples informations, veuillez consulter [Afficher les CodeBuild métriques](#).

## CodeBuild CloudWatch métriques

Les indicateurs suivants peuvent être suivis par AWS compte ou par projet de création. Pour plus d'informations sur l'utilisation CloudWatch avec CodeBuild, consultez [Surveillez CodeBuild les builds avec CloudWatch](#).

### BuildDuration

Mesure la durée de la phase BUILD de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

### Builds

Mesure le nombre de générations déclenchées.

Unités : nombre

CloudWatch Statistiques valides : Sum

### DownloadSourceDuration

Mesure la durée de la phase DOWNLOAD\_SOURCE de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

### Duration

Mesure la durée de toutes les générations au fil du temps.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## FailedBuilds

Mesure le nombre de builds qui ont échoué en raison d'une erreur du client ou d'un délai d'attente.

Unités : nombre

CloudWatch Statistiques valides : Sum

## FinalizingDuration

Mesure la durée de la phase FINALIZING de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## InstallDuration

Mesure la durée de la phase INSTALL de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## PostBuildDuration

Mesure la durée de la phase POST\_BUILD de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## PreBuildDuration

Mesure la durée de la phase PRE\_BUILD de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## ProvisioningDuration

Mesure la durée de la phase PROVISIONING de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## QueuedDuration

Mesure la durée de la phase QUEUED de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## SubmittedDuration

Mesure la durée de la phase SUBMITTED de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## SucceededBuilds

Mesure le nombre de générations réussies.

Unités : nombre

CloudWatch Statistiques valides : Sum

## UploadArtifactsDuration

Mesure la durée de la phase UPLOAD\_ARTIFACTS de la génération.

Unités : secondes

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## CodeBuild CloudWatch mesures d'utilisation des ressources

### Note

CodeBuild les mesures d'utilisation des ressources ne sont disponibles que dans les régions suivantes :

- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Asie-Pacifique (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asia Pacific (Sydney)

- Région Canada (Centre)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région Amérique du Sud (São Paulo)
- US East (N. Virginia) Region
- US East (Ohio) Region
- Région US West (N. California)
- Région US West (Oregon)

Les mesures d'utilisation des ressources suivantes peuvent être suivies. Pour plus d'informations sur l'utilisation CloudWatch avec CodeBuild, consultez [Surveillez CodeBuild les builds avec CloudWatch](#).

#### CPUUtilized

Le nombre d'unités de processeur allouées au traitement utilisé par le conteneur de construction.

Unités : unités CPU

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

#### CPUUtilizedPourcentage

Pourcentage de traitement alloué utilisé par le conteneur de construction.

Unités : pourcentage

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

#### MemoryUtilized

Nombre de mégaoctets de mémoire utilisés par le conteneur de compilation.

Unités : mégaoctets

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

#### MemoryUtilizedPercent

Pourcentage de mémoire allouée utilisé par le conteneur de compilation.

Unités : pourcentage

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

StorageReadBytes

Vitesse de lecture du stockage utilisée par le conteneur de construction.

Unités : octets/seconde

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

StorageWriteBytes

Vitesse d'écriture de stockage utilisée par le conteneur de construction.

Unités : octets/seconde

CloudWatch Statistiques valides : moyenne (recommandée), maximale, minimale

## CodeBuild CloudWatch dimensions

CodeBuild fournit les dimensions CloudWatch métriques suivantes. Si aucune de ces valeurs n'est spécifiée, les mesures concernent le AWS compte courant.

BuildId, BuildNumber, ProjectName

Des métriques sont fournies pour un identifiant de build, un numéro de build et un nom de projet.

ProjectName

Des métriques sont fournies pour le nom d'un projet.

## CodeBuild CloudWatch alarmes

Vous pouvez utiliser la CloudWatch console pour créer des alarmes basées sur CodeBuild des métriques afin de pouvoir réagir en cas de problème avec vos builds. Les deux mesures les plus utiles avec les alarmes sont décrites dans les points suivants. Pour plus d'informations sur l'utilisation CloudWatch avec CodeBuild, consultez [Surveillez CodeBuild les builds avec CloudWatch](#).

- **FailedBuild.** Vous pouvez créer une alarme qui se déclenche lorsqu'un certain nombre de builds échoués sont détectés dans un délai prédéterminé de secondes. Dans CloudWatch, vous spécifiez le nombre de secondes et le nombre de builds échoués qui déclenchent une alarme.

- **Duration.** Vous pouvez créer une alarme qui se déclenche lorsqu'une construction prend plus de temps que prévu. Vous spécifiez le nombre de secondes qui doit s'écouler entre le démarrage et la fin d'une génération, avant le déclenchement de l'alarme.

Pour plus d'informations sur la création d'alarmes pour CodeBuild les métriques, consultez [Surveillez les CodeBuild builds avec des CloudWatch alarmes](#). Pour plus d'informations sur les alarmes, consultez la section [Création d' CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon.

## Afficher les CodeBuild métriques

AWS CodeBuild surveille les fonctions en votre nom et publie des statistiques via Amazon CloudWatch. Ces métriques incluent le nombre total de générations, d'échecs et de réussites, ainsi que la durée des générations.

Vous pouvez utiliser la CodeBuild console ou la CloudWatch console pour surveiller les métriques de CodeBuild. Les procédures suivantes vous montrent comment afficher les métriques.

### Rubriques

- [Afficher les statistiques de construction \(CodeBuild console\)](#)
- [Afficher les statistiques de construction \( CloudWatch console Amazon\)](#)

## Afficher les statistiques de construction (CodeBuild console)

### Note

Vous ne pouvez pas personnaliser les statistiques ou les graphiques utilisés pour les afficher dans la CodeBuild console. Si vous souhaitez personnaliser l'affichage, utilisez la CloudWatch console Amazon pour consulter les statistiques de votre build.

### Indicateurs au niveau du compte

Pour consulter les statistiques au AWS niveau du compte

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.



2. Dans le volet de navigation, sélectionnez Métriques de compte.

## Métriques au niveau du projet

Pour afficher les métriques au niveau du projet

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de génération, dans la colonne Nom, choisissez le projet dont vous souhaitez consulter les métriques.
4. Sélectionnez l'onglet Métriques.

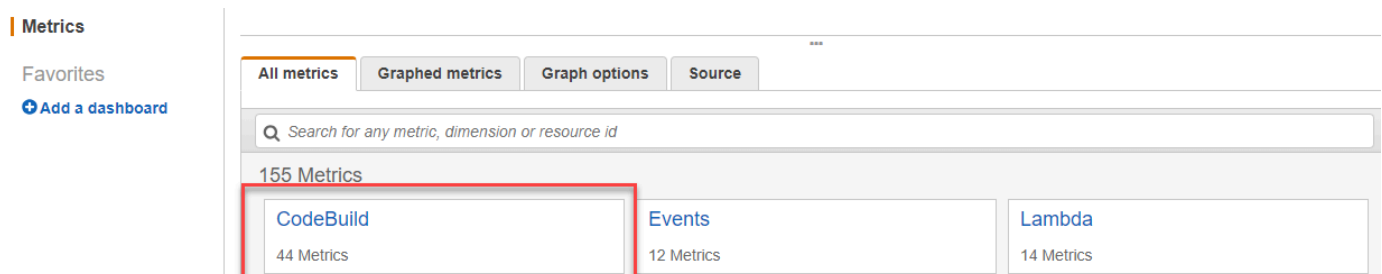
## Afficher les statistiques de construction ( CloudWatch console Amazon)

Vous pouvez personnaliser les métriques et les graphiques utilisés pour les afficher avec la CloudWatch console.

## Indicateurs au niveau du compte

Pour consulter les statistiques au niveau du compte

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sous l'onglet All metrics (Toutes les métriques), choisissez CodeBuild.

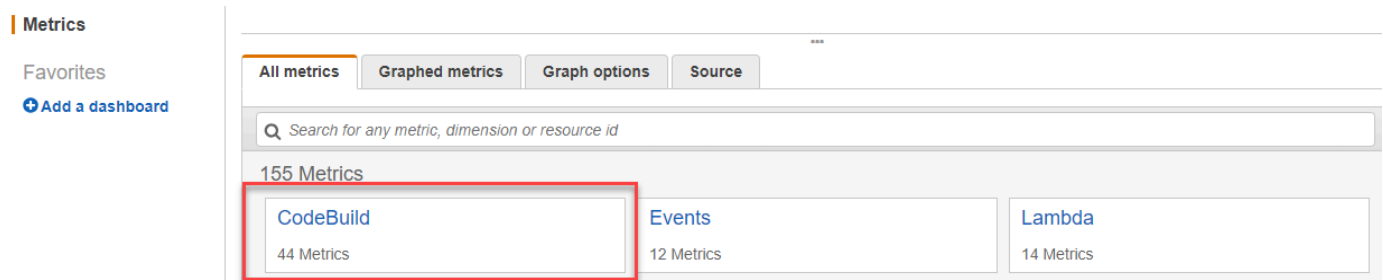


4. Choisissez Account metrics (Métriques globales).
5. Choisissez un ou plusieurs projets et métriques. Pour chaque projet, vous pouvez choisir les métriques SucceededBuildsFailedBuilds, Builds et Duration. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.

## Métriques au niveau du projet

Pour afficher les métriques au niveau du projet

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sous l'onglet All metrics (Toutes les métriques), choisissez CodeBuild.



4. Choisissez Par projet.
5. Choisissez une ou plusieurs combinaisons de projets et de métriques. Pour chaque projet, vous pouvez choisir les métriques SucceededBuildsFailedBuilds, Builds et Duration. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.
6. (Facultatif) Vous pouvez personnaliser vos métriques et graphiques. Par exemple, dans la liste déroulante de la colonne Statistiques, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour plus d'informations, consultez [Graph metrics](#) et [View available metrics](#) dans le Amazon CloudWatch User Guide.

## Afficher les indicateurs d'utilisation des CodeBuild ressources

AWS CodeBuild surveille l'utilisation des ressources de construction en votre nom et publie des statistiques via Amazon CloudWatch. Il s'agit notamment de mesures telles que l'utilisation du processeur, de la mémoire et du stockage.

 Note

CodeBuild les mesures d'utilisation des ressources ne sont enregistrées que pour les versions qui s'exécutent pendant plus d'une minute.

Vous pouvez utiliser la CodeBuild console ou la CloudWatch console pour surveiller les mesures d'utilisation des ressources pour CodeBuild.

 Note

CodeBuild les mesures d'utilisation des ressources ne sont disponibles que dans les régions suivantes :

- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Asie-Pacifique (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asia Pacific (Sydney)
- Région Canada (Centre)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région Amérique du Sud (São Paulo)
- US East (N. Virginia) Region
- US East (Ohio) Region
- Région US West (N. California)
- Région US West (Oregon)

Les procédures suivantes vous indiquent comment accéder à vos indicateurs d'utilisation des ressources.

## Rubriques

- [Accédez aux métriques d'utilisation des ressources \(CodeBuild console\)](#)
- [Accédez aux métriques d'utilisation des ressources \( CloudWatch console Amazon\)](#)

## Accédez aux métriques d'utilisation des ressources (CodeBuild console)

### Note

Vous ne pouvez pas personnaliser les statistiques ou les graphiques utilisés pour les afficher dans la CodeBuild console. Si vous souhaitez personnaliser l'affichage, utilisez la CloudWatch console Amazon pour consulter les statistiques de votre build.

### Mesures d'utilisation des ressources au niveau du projet

Pour accéder aux indicateurs d'utilisation des ressources au niveau du projet

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de construction, dans la colonne Nom, choisissez le projet pour lequel vous souhaitez consulter les mesures d'utilisation.
4. Sélectionnez l'onglet Métriques. Les mesures d'utilisation des ressources sont affichées dans la section Mesures d'utilisation des ressources.
5. Pour afficher les mesures d'utilisation des ressources au niveau du projet dans la CloudWatch console, choisissez Afficher CloudWatch dans la section Mesures d'utilisation des ressources.

### Mesures d'utilisation des ressources au niveau du build

Pour accéder aux indicateurs d'utilisation des ressources au niveau du build

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la sur <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Historique de génération.
3. Dans la liste des versions, dans la colonne Build run, choisissez la version pour laquelle vous souhaitez afficher les métriques d'utilisation.
4. Choisissez l'onglet Utilisation des ressources.

5. Pour afficher les mesures d'utilisation des ressources au niveau du build dans la CloudWatch console, choisissez Afficher CloudWatch dans la section Mesures d'utilisation des ressources.

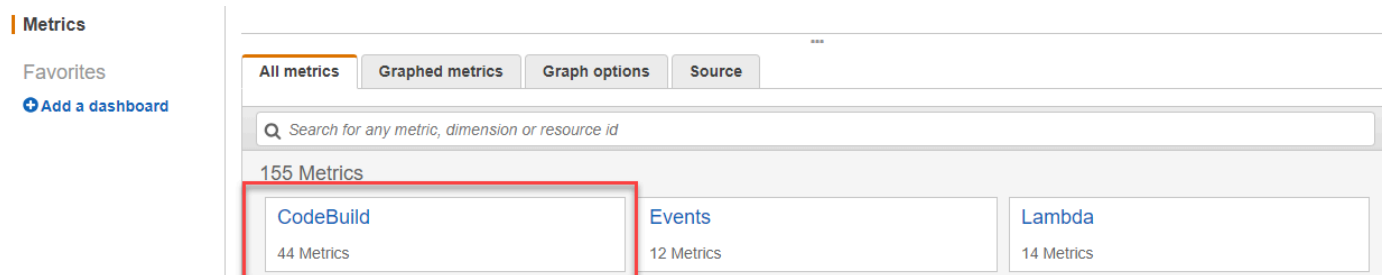
## Accédez aux métriques d'utilisation des ressources ( CloudWatch console Amazon)

La CloudWatch console Amazon peut être utilisée pour accéder aux métriques d'utilisation des CodeBuild ressources.

### Mesures d'utilisation des ressources au niveau du projet

Pour accéder aux indicateurs d'utilisation des ressources au niveau du projet

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sous l'onglet All metrics (Toutes les métriques), choisissez CodeBuild.



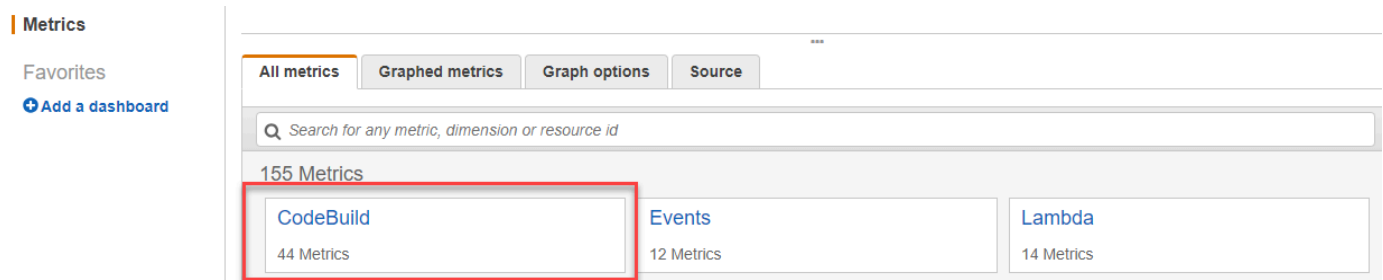
4. Choisissez Par projet.
5. Choisissez une ou plusieurs combinaisons de projets et de métriques à ajouter au graphique. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.
6. (Facultatif) Vous pouvez personnaliser vos statistiques et vos graphiques à partir de l'onglet Mesures graphiques. Par exemple, dans la liste déroulante de la colonne Statistiques, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour plus d'informations, consultez les sections [Représentation graphique des métriques](#) et [Affichage des métriques disponibles](#) dans le guide de CloudWatch l'utilisateur Amazon.

## Mesures d'utilisation des ressources au niveau du build

Pour accéder aux indicateurs d'utilisation des ressources au niveau du build

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sous l'onglet All metrics (Toutes les métriques), choisissez CodeBuild.



4. Choisissez BuildId BuildNumber, ProjectName.
5. Choisissez une ou plusieurs combinaisons de constructions et de métriques à ajouter au graphique. Toutes les combinaisons de construction et de métriques sélectionnées sont affichées dans le graphique de la page.
6. (Facultatif) Vous pouvez personnaliser vos statistiques et vos graphiques à partir de l'onglet Mesures graphiques. Par exemple, dans la liste déroulante de la colonne Statistiques, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour plus d'informations, consultez les sections [Représentation graphique des métriques](#) et [Affichage des métriques disponibles](#) dans le guide de CloudWatch l'utilisateur Amazon.

## Surveillez les CodeBuild builds avec des CloudWatch alarmes

Vous pouvez créer une CloudWatch alarme pour vos builds. Une alarme surveille une métrique individuelle pendant une période que vous définissez et exécute une ou plusieurs actions en fonction de la valeur de cette métrique, par rapport à un seuil spécifié sur un certain nombre de périodes. À l'aide de la fonctionnalité CloudWatch d'alarme native, vous pouvez spécifier toutes les actions prises en charge en CloudWatch cas de dépassement d'un seuil. Par exemple, vous pouvez spécifier qu'une notification Amazon SNS est envoyée lorsque plus de trois versions de votre compte échouent dans les quinze minutes.

## Pour créer une CloudWatch alarme pour une CodeBuild métrique

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, cliquez sur Alarms (Alarmes).
3. Sélectionnez Create Alarm (Créer une alarme).
4. Sous CloudWatch Métriques par catégorie, sélectionnez CodeBuildMétriques. Si vous savez que vous utiliserez uniquement les métriques de niveau projet, choisissez Par projet. Si vous savez que vous utiliserez uniquement les métriques de niveau compte, choisissez Métriques de compte.
5. Sous Créer une alarme, si l'option n'est pas déjà sélectionnée, choisissez Sélectionner une métrique.
6. Choisissez la métrique pour laquelle vous voulez créer une alarme. Les options disponibles sont Par projet et Métriques de compte.
7. Choisissez Suivant ou Définir une alarme, puis créez votre alarme. Pour plus d'informations, consultez la section [Création d' CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon. Pour plus d'informations sur la configuration des notifications Amazon SNS lorsqu'une alarme est déclenchée, consultez la section [Configurer les notifications Amazon SNS](#) dans le manuel du développeur Amazon SNS.
8. Sélectionnez Create Alarm (Créer une alerte).

# Sécurité dans AWS CodeBuild

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- **Sécurité du cloud** : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des [programmes de conformité AWS](#). Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS CodeBuild, consultez la section [AWS Services concernés par programme de conformité](#).
- **Sécurité dans le cloud** — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation CodeBuild. Les rubriques suivantes expliquent comment procéder à la configuration CodeBuild pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos CodeBuild ressources.

## Rubriques

- [Protection des données dans AWS CodeBuild](#)
- [Gestion des identités et des accès dans AWS CodeBuild](#)
- [Validation de conformité pour AWS CodeBuild](#)
- [Résilience dans AWS CodeBuild](#)
- [Sécurité de l'infrastructure dans AWS CodeBuild](#)
- [Accédez à votre fournisseur source dans CodeBuild](#)
- [Prévention du cas de figure de l'adjoint désorienté entre services](#)



# Protection des données dans AWS CodeBuild

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS CodeBuild. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée d'AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez [Norme FIPS \(Federal Information Processing Standard\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec CodeBuild ou d'autres Services AWS

utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Pour protéger les informations sensibles, les informations suivantes sont masquées dans CodeBuild les journaux :

- Chaînes spécifiées à l'aide du Parameter Store dans les variables d'environnement CodeBuild du projet ou dans la section `buildspecenv/parameter-store`. Pour plus d'informations, consultez la [présentation de la console Systems Manager Parameter Store et Systems Manager Parameter Store dans le guide](#) de l'utilisateur d'Amazon EC2 Systems Manager.
- Chaînes spécifiées AWS Secrets Manager à l'aide des variables d'environnement CodeBuild du projet ou de la section `buildspecenv/secrets-manager`. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

Pour de plus amples informations sur la protection des données, veuillez consulter le billet de blog [Modèle de responsabilité partagée AWS et RGPD](#) à la page Blog sur la sécuritéAWS .

## Rubriques

- [Chiffrement des données](#)
- [Gestion des clés](#)
- [Confidentialité du trafic](#)

## Chiffrement des données

Le chiffrement est un élément important de la CodeBuild sécurité. Certains chiffrements, par exemple pour les données en transit, sont fournis par défaut et vous n'avez pas besoin de faire quoi que ce soit. Vous pouvez en configurer d'autres, par exemple pour les données au repos, lorsque vous créez votre projet ou votre génération.

- Chiffrement des données au repos - Les artefacts de build, tels qu'un cache, des journaux, des fichiers de données de rapports de test bruts exportés et des résultats de build, sont chiffrés par défaut à l'aide Clés gérées par AWS de. Si vous ne souhaitez pas utiliser ces clés KMS, vous devez créer et configurer une clé gérée par le client. Pour de plus amples informations, veuillez

consulter [Création de clés KMS](#) et [Concepts d'AWS Key Management Service](#) dans le Guide de l'utilisateur AWS Key Management Service .

- Vous pouvez stocker l'identifiant de la clé AWS KMS CodeBuild utilisée pour chiffrer l'artefact de sortie de construction dans la variable d'CODEBUILD\_KMS\_KEY\_IDenvironnement. Pour de plus amples informations, consultez [Variables d'environnement dans les environnements de génération](#).
- Vous pouvez spécifier une clé gérée par le client lorsque vous créez un projet de construction. Pour plus d'informations, consultez [Set the Encryption Key Using the Console](#) et [Définissez la clé de chiffrement à l'aide de la CLI](#).

Les volumes Amazon Elastic Block Store de votre parc de produits sont chiffrés par défaut à l'aide de Clés gérées par AWS.

- Chiffrement des données en transit : toutes les communications entre les clients CodeBuild et entre CodeBuild et leurs dépendances en aval sont protégées par des connexions TLS signées à l'aide du processus de signature Signature version 4. Tous les CodeBuild points de terminaison utilisent des certificats SHA-256 gérés par AWS Autorité de certification privée Pour de plus amples informations, veuillez consulter [Processus de signature Signature Version 4](#) et [Présentation d'ACM PCA](#).
- Chiffrement des artefacts de génération : le rôle de CodeBuild service associé au projet de génération nécessite l'accès à une clé KMS afin de chiffrer ses artefacts de sortie de génération. Par défaut, CodeBuild utilise un Clé gérée par AWS pour Amazon S3 dans votre AWS compte. Si vous ne souhaitez pas l'utiliser Clé gérée par AWS, vous devez créer et configurer une clé gérée par le client. Pour plus d'informations, reportez-vous à [Chiffrer les sorties de build](#) la section [Création de clés](#) dans le Guide du AWS KMS développeur.

## Gestion des clés

Vous pouvez protéger votre contenu d'une utilisation non autorisée grâce au chiffrement. Stockez vos clés de chiffrement dans AWS Secrets Manager, puis autorisez le rôle de CodeBuild service associé au projet de génération à obtenir les clés de chiffrement depuis votre compte Secrets Manager. Pour de plus amples informations, veuillez consulter [Chiffrez les résultats de construction à l'aide d'une clé gérée par le client](#), [Créez un projet de construction dans AWS CodeBuild](#), [Exécuter AWS CodeBuild les builds manuellement](#) et [Didacticiel : Stockage et récupération d'un secret](#).

Utilisez la variable d'environment `CODEBUILD_KMS_KEY_ID` dans une commande de construction pour obtenir l'identifiant de AWS KMS clé. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

Vous pouvez utiliser Secrets Manager pour protéger les informations d'identification d'un registre privé qui stocke une image Docker utilisée pour votre environnement d'exécution. Pour de plus amples informations, veuillez consulter [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#).

## Confidentialité du trafic

Vous pouvez améliorer la sécurité de vos builds en les configurant CodeBuild pour utiliser un point de terminaison VPC d'interface. Pour ce faire, vous n'avez pas besoin d'une passerelle Internet, d'un périphérique NAT ni d'une passerelle privée virtuelle. Il n'est pas non plus nécessaire de le configurer PrivateLink, bien que cela soit recommandé. Pour de plus amples informations, veuillez consulter [Utilisation de points de terminaison de VPC](#). Pour plus d'informations sur PrivateLink les points de terminaison VPC, consultez [AWS PrivateLink](#) et [Accès aux AWS](#) services via. PrivateLink

## Gestion des identités et des accès dans AWS CodeBuild

L'accès à AWS CodeBuild nécessite des informations d'identification. Ces informations d'identification doivent être autorisées à accéder aux AWS ressources, telles que le stockage et la récupération d'artefacts de build dans des compartiments S3 et la consultation d'Amazon CloudWatch Logs pour les builds. Les sections suivantes décrivent comment vous pouvez utiliser [Gestion des identités et des accès AWS](#)(IAM) et comment CodeBuild sécuriser l'accès à vos ressources :

### Vue d'ensemble de la gestion des autorisations d'accès à vos AWS CodeBuild ressources

Chaque AWS ressource appartient à un AWS compte, et les autorisations de création ou d'accès à une ressource sont régies par des politiques d'autorisation. Un compte administrateur peut attacher des politiques d'autorisations à des identités IAM (c'est-à-dire des utilisateurs, des groupes et des rôles).

**Note**

Un administrateur de compte (ou utilisateur administrateur) est un utilisateur doté des privilèges d'administrateur. Pour plus d'informations, consultez [Bonnes pratiques IAM](#) dans le Guide de l'utilisateur IAM.

Lorsque vous accordez des autorisations, vous décidez qui obtient les autorisations, les ressources auxquelles elles peuvent accéder et les actions pouvant être effectuées sur ces ressources.

**Rubriques**

- [AWS CodeBuild ressources et opérations](#)
- [Présentation de la propriété des ressources](#)
- [Gestion de l'accès aux ressources](#)
- [Spécification des éléments d'une politique : actions, effets et mandataires](#)

**AWS CodeBuild ressources et opérations**

Dans AWS CodeBuild, la ressource principale est un projet de construction. Dans une stratégie, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la stratégie s'applique. Les builds sont également des ressources auxquelles ils sont ARNs associés. Pour plus d'informations, consultez [Amazon Resource Names \(ARN\) et AWS Service Namespaces](#) dans le Référence générale d'Amazon Web Services.

| Type de ressource    | Format ARN                                                                                      |
|----------------------|-------------------------------------------------------------------------------------------------|
| Génération de projet | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :project/ <i>project-name</i>           |
| Génération           | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :build/ <i>build-ID</i>                 |
| Groupe de rapports   | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report-group/ <i>report-group-name</i> |
| Rapport              | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report/ <i>report-ID</i>               |

| Type de ressource                                                                            | Format ARN                                                                                 |
|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Flotte                                                                                       | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :fleet/<i>fleet-ID</i></code> |
| Toutes les CodeBuild ressources                                                              | <code>arn:aws:codebuild:*</code>                                                           |
| Toutes les CodeBuild ressources détenues par le compte spécifié dans la AWS région spécifiée | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :*</code>                     |

#### Important

Lorsque vous utilisez la fonctionnalité de capacité réservée, les données mises en cache sur les instances de flotte, y compris les fichiers source, les couches Docker et les répertoires mis en cache spécifiés dans les spécifications de construction, peuvent être accessibles aux autres projets du même compte. Cela est intentionnel et permet aux projets au sein d'un même compte de partager des instances de flotte.

#### Note

La plupart AWS des services considèrent les deux points (:)) ou les barres obliques (/) comme le même caractère dans ARNs. Cependant, CodeBuild utilise une correspondance exacte dans les modèles de ressources et les règles. Veillez à utiliser les caractères corrects lors de la création de modèles d'événements, afin qu'ils correspondent à la syntaxe ARN de la ressource.

Par exemple, vous pouvez indiquer un projet de construction spécifique (*myBuildProject*) dans votre instruction en utilisant son ARN comme suit :

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Pour spécifier toutes les ressources, ou si une action d'API n'est pas compatible ARNs, utilisez le caractère générique (\*) dans l'Resource élément comme suit :

```
"Resource": "*"
```

Certaines actions CodeBuild d'API acceptent plusieurs ressources (par exemple, `BatchGetProjects`). Pour spécifier plusieurs ressources dans une seule instruction, séparez-les ARNs par des virgules, comme suit :

```
"Resource": [
 "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",
 "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"
]
```

CodeBuild fournit un ensemble d'opérations permettant de travailler avec les CodeBuild ressources. Pour en obtenir la liste, consultez [AWS CodeBuild référence aux autorisations](#).

## Présentation de la propriété des ressources

Le AWS compte possède les ressources créées dans le compte, quelle que soit la personne qui les a créées. Plus précisément, le propriétaire de la ressource est le AWS compte de l'[entité principale](#) (c'est-à-dire le compte root, un utilisateur ou un rôle IAM) qui authentifie la demande de création de ressource. Les exemples suivants illustrent comment cela fonctionne :

- Si vous utilisez les informations d'identification du compte root de votre AWS compte pour créer une règle, votre AWS compte est le propriétaire de la CodeBuild ressource.
- Si vous créez un utilisateur dans votre AWS compte et que vous accordez l'autorisation de créer CodeBuild des ressources à cet utilisateur, celui-ci peut créer CodeBuild des ressources. Cependant, votre AWS compte, auquel appartient l'utilisateur, est propriétaire des CodeBuild ressources.
- Si vous créez un rôle IAM dans votre AWS compte avec les autorisations nécessaires pour créer CodeBuild des ressources, toute personne pouvant assumer ce rôle peut créer des CodeBuild ressources. Votre AWS compte, auquel appartient le rôle, est propriétaire des CodeBuild ressources.

## Gestion de l'accès aux ressources

Une politique d'autorisations décrit qui a accès à quelles ressources.

 Note

Cette section décrit l'utilisation d'IAM dans AWS CodeBuild. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, consultez [Qu'est-ce que IAM ?](#) dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des stratégies IAM, consultez [Référence de stratégie AWS IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques attachées à une identité IAM sont appelées des politiques basées sur l'identité (politiques IAM). Les politiques associées à une ressource sont appelées politiques basées sur les ressources. CodeBuild prend en charge les politiques basées sur l'identité et les politiques basées sur les ressources pour certaines applications en lecture seule à des fins de APIs partage de ressources entre comptes.

### Accès sécurisé aux compartiments S3

Nous vous recommandons vivement d'inclure les autorisations suivantes dans votre rôle IAM afin de vérifier que le compartiment S3 associé à votre CodeBuild projet est votre propriété ou celle d'une personne de confiance. Ces autorisations ne sont pas incluses dans les politiques et les rôles AWS gérés. Vous devez les ajouter vous-même.

- `s3:GetBucketAc1`
- `s3:GetBucketLocation`

Si le propriétaire d'un compartiment S3 utilisé par votre projet change, vous devez vérifier que vous êtes toujours propriétaire du compartiment et mettre à jour les autorisations dans votre rôle IAM dans le cas contraire. Pour plus d'informations, consultez [Permettre aux utilisateurs d'interagir avec CodeBuild](#) et [CodeBuild Autoriser l'interaction avec d'autres AWS services](#).

### Spécification des éléments d'une politique : actions, effets et mandataires

Pour chaque AWS CodeBuild ressource, le service définit un ensemble d'opérations d'API. Pour accorder des autorisations pour ces opérations d'API CodeBuild , définissez un ensemble d'actions que vous pouvez spécifier dans une politique. Certaines opérations d'API peuvent exiger des autorisations pour plusieurs actions afin de réaliser l'opération d'API. Pour plus d'informations, consultez [AWS CodeBuild ressources et opérations](#) et [AWS CodeBuild référence aux autorisations](#).

Voici les éléments de base d'une politique :



- **Ressource** : vous utilisez un nom Amazon Resource Name (ARN) pour identifier la ressource à laquelle s'applique la politique.
- **Action** : vous utilisez des mots clés d'action pour identifier les opérations sur les ressources que vous souhaitez autoriser ou refuser. Par exemple, l'autorisation `codebuild:CreateProject` autorise l'utilisateur à effectuer l'opération `CreateProject`.
- **Effet** : vous spécifiez l'effet, qu'il s'agisse d'autoriser ou de refuser, lorsque l'utilisateur demande l'action. Si vous n'accordez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez également explicitement refuser l'accès à une ressource. Vous pouvez le faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une stratégie différente accorde cet accès.
- **Principal** — Dans les politiques basées sur l'identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur une ressource, vous spécifiez l'utilisateur, le compte, le service ou une autre entité qui doit recevoir les autorisations.

Pour en savoir plus sur la syntaxe des stratégies IAM et pour obtenir des descriptions, consultez [Référence de stratégie IAM AWS](#) dans le Guide de l'utilisateur IAM.

Pour un tableau présentant toutes les actions d' CodeBuild API et les ressources auxquelles elles s'appliquent, consultez le [AWS CodeBuild référence aux autorisations](#).

## Utilisation de politiques basées sur l'identité pour AWS CodeBuild

Cette rubrique fournit des exemples de stratégies basées sur l'identité qui montrent comment un administrateur de compte peut lier des stratégies d'autorisations à des identités IAM (autrement dit, des utilisateurs, des groupes et des rôles) et accorder ainsi des autorisations pour effectuer des opérations sur les ressources AWS CodeBuild .

### Important

Nous vous recommandons de consulter d'abord les rubriques d'introduction qui expliquent les concepts de base et les options disponibles pour gérer l'accès à vos CodeBuild ressources. Pour de plus amples informations, veuillez consulter [Vue d'ensemble de la gestion des autorisations d'accès à vos AWS CodeBuild ressources](#).

## Rubriques

- [Autorisations requises pour utiliser la AWS CodeBuild console](#)
- [Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry](#)
- [Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources](#)
- [AWS politiques gérées \(prédéfinies\) pour AWS CodeBuild](#)
- [CodeBuild politiques et notifications gérées](#)
- [CodeBuild mises à jour des politiques AWS gérées](#)
- [Exemples de politiques gérées par le client](#)

Voici un exemple de stratégie d'autorisations qui autorise un utilisateur à obtenir des informations sur des projets de génération uniquement dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetProjects",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

## Autorisations requises pour utiliser la AWS CodeBuild console

Un utilisateur qui utilise la AWS CodeBuild console doit disposer d'un ensemble minimal d'autorisations lui permettant de décrire les autres AWS ressources du AWS compte. Vous devez détenir les autorisations des services suivants :

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (si vous stockez votre code source dans un AWS CodeCommit dépôt)

- Amazon Elastic Container Registry (Amazon ECR) (si vous utilisez un environnement de construction qui repose sur une image Docker dans un référentiel Amazon ECR)

 Note

Depuis le 26 juillet 2022, la politique IAM par défaut a été mise à jour. Pour de plus amples informations, veuillez consulter [Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (si vous utilisez un environnement de construction qui repose sur une image Docker dans un référentiel Amazon ECR)
- Gestion des identités et des accès AWS (JE SUIS)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu.

## Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry

Depuis le 26 juillet 2022, AWS CodeBuild a mis à jour sa politique IAM par défaut pour les autorisations Amazon ECR. Les autorisations suivantes ont été supprimées de la politique par défaut :

```
"ecr:PutImage",
"ecr:InitiateLayerUpload",
"ecr:UploadLayerPart",
"ecr:CompleteLayerUpload"
```

Pour les CodeBuild projets créés avant le 26 juillet 2022, nous vous recommandons de mettre à jour votre politique avec la politique Amazon ECR suivante :

```
"Action": [
 "ecr:BatchCheckLayerAvailability",
 "ecr:GetDownloadUrlForLayer",
 "ecr:BatchGetImage"
```

]

Pour plus d'informations sur la mise à jour de votre politique, consultez [Permettre aux utilisateurs d'interagir avec CodeBuild](#).

## Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources

La AWS CodeBuild console utilise les actions d'API suivantes pour se connecter aux fournisseurs de sources (par exemple, les GitHub référentiels).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Vous pouvez associer des fournisseurs de sources (tels que GitHub des référentiels) à vos projets de génération à l'aide de la AWS CodeBuild console. Pour ce faire, vous devez d'abord ajouter les actions d'API précédentes aux politiques d'accès IAM associées à l'utilisateur que vous utilisez pour accéder à la AWS CodeBuild console.

Les actions d'API `ListConnectedOAuthAccounts`, `ListRepositories` et `PersistOAuthToken` ne sont pas conçues pour être appelées par votre code. Par conséquent, ces actions d'API ne sont pas incluses dans le AWS CLI et AWS SDKs.

## AWS politiques gérées (prédéfinies) pour AWS CodeBuild

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par AWS. Ces politiques AWS gérées accordent les autorisations nécessaires pour les cas d'utilisation courants afin que vous n'ayez pas à rechercher les autorisations nécessaires. Les politiques gérées fournissent CodeBuild également des autorisations pour effectuer des opérations dans d'autres services, tels que IAM AWS CodeCommit, Amazon EC2, Amazon ECR, Amazon SNS et CloudWatch Amazon Events, conformément aux responsabilités des utilisateurs auxquels la politique en question a été accordée. Par exemple, il s'agit d'une `AWSCodeBuildAdminAccess` politique utilisateur de niveau administratif qui permet aux utilisateurs dotés de cette politique de créer et de gérer des règles relatives aux CloudWatch événements pour les versions de projets et des rubriques Amazon SNS pour les notifications relatives aux

événements liés au projet (sujets dont les noms sont préfixés par `arn:aws:codebuild:`), ainsi que d'administrer des projets et des groupes de rapports dans CodeBuild. Pour plus d'informations, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à AWS CodeBuild.

#### `AWSCodeBuildAdminAccess`

Fournit un accès complet CodeBuild, y compris les autorisations pour administrer les projets de CodeBuild construction.

#### `AWSCodeBuildDeveloperAccess`

Permet d'accéder à l'administration du projet de build, CodeBuild mais ne l'autorise pas.

#### `AWSCodeBuildReadOnlyAccess`

Fournit un accès en lecture seule à CodeBuild.

Pour accéder aux artefacts de sortie de build CodeBuild créés, vous devez également joindre la politique AWS gérée nommée `AmazonS3ReadOnlyAccess`.

Pour créer et gérer des rôles de CodeBuild service, vous devez également associer la politique AWS gérée nommée `IAMFullAccess`.

Vous pouvez également créer vos propres politiques IAM personnalisées afin d'accorder des autorisations pour les actions et les ressources CodeBuild. Vous pouvez attacher ces stratégies personnalisées aux utilisateurs ou groupes qui nécessitent ces autorisations.

#### Rubriques

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

#### `AWSCodeBuildAdminAccess`

La `AWSCodeBuildAdminAccess` politique fournit un accès complet aux projets de CodeBuild construction CodeBuild, y compris les autorisations nécessaires à leur administration. Appliquez cette

politique uniquement aux utilisateurs de niveau administratif afin de leur accorder un contrôle total sur les CodeBuild projets, les groupes de rapports et les ressources associées de votre AWS compte, y compris la possibilité de supprimer des projets et des groupes de rapports.

Pour la stratégie entièrement gérée, voir la référence [AWSCodeBuildAdminAccess](#) des politiques gérées par IAM.

#### AWSCodeBuildDeveloperAccess

La `AWSCodeBuildDeveloperAccess` politique permet d'accéder à toutes les fonctionnalités et aux ressources liées aux projets CodeBuild et aux groupes de rapports. Cette politique n'autorise pas les utilisateurs à supprimer CodeBuild des projets, des groupes de rapports ou des ressources connexes dans d'autres AWS services, tels que CloudWatch les événements. Nous vous recommandons d'appliquer cette stratégie à la plupart des utilisateurs.

Pour la stratégie entièrement gérée, voir la référence [AWSCodeBuildDeveloperAccess](#) des politiques gérées par IAM.

#### AWSCodeBuildReadOnlyAccess

La `AWSCodeBuildReadOnlyAccess` politique accorde un accès en lecture seule aux CodeBuild ressources associées dans d'autres AWS services. Appliquez cette stratégie aux utilisateurs qui peuvent afficher et exécuter des builds, afficher des projets et afficher des groupes de rapports, mais qui ne peuvent pas y apporter de modifications.

Pour la stratégie entièrement gérée, voir la référence [AWSCodeBuildReadOnlyAccess](#) des politiques gérées par IAM.

## CodeBuild politiques et notifications gérées

CodeBuild prend en charge les notifications, qui peuvent informer les utilisateurs des modifications importantes apportées aux projets de construction. Les politiques gérées CodeBuild incluent des déclarations de politique relatives à la fonctionnalité de notification. Pour plus d'informations, consultez [En quoi consistent les notifications ?](#)

### Autorisations liées aux notifications dans les stratégies gérées en lecture seule

La stratégie gérée `AWSCodeBuildReadOnlyAccess` inclut les déclarations suivantes pour autoriser l'accès en lecture seule aux notifications. Les utilisateurs auxquels s'applique cette stratégie gérée

peuvent voir des notifications pour les ressources, mais ne peuvent ni les créer, ni les gérer ni s'y abonner.

```
{
 "Sid": "CodeStarNotificationsPowerUserAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:DescribeNotificationRule"
],
 "Resource": "*",
 "Condition" : {
 "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets"
],
 "Resource": "*"
}
```

### Autorisations liées aux notifications dans d'autres stratégies gérées

La stratégie gérée `AWSCodeBuildDeveloperAccess` inclut les déclarations suivantes pour autoriser les utilisateurs à créer et modifier des notifications, et s'y abonner. Les utilisateurs ne peuvent pas supprimer les règles de notification ni gérer les balises pour les ressources.

```
{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
}
```

```

 "Condition" : {
 "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
 },
 {
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource",
 "codestar-notifications:ListEventTypes"
],
 "Resource": "*"
 },
 {
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
 }
]
}

```

Pour plus d'informations sur l'IAM et les notifications, voir [Identity and Access ManagementAWS CodeStarNotifications](#).

## CodeBuild mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CodeBuild depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au fil RSS sur [AWS CodeBuild Historique du document du guide de l'utilisateur](#).



| Modifier                                                                                                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Date                     |
|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <p>AWSCodeBuildAdminAccess et AWSCodeBuildDeveloperAccess — Mise à jour des politiques existantes</p>                               | <p>CodeBuild a ajouté l'ssmessages:OpenDataChannel autorisation à ces politiques pour prendre en charge le débogage interactif des versions de Session Manager.</p> <p>Les AWSCodeBuildDeveloperAccess politiques AWSCodeBuildAdminAccess et incluent désormais l'ssmessages:OpenDataChannel action permettant aux ressources de session du gestionnaire de session (arn:aws:ssm:*:*:session/* ) de prendre en charge l'application du protocole SigV4 sur cette WebSocket API.</p> | <p>1er décembre 2025</p> |
| <p>AWSCodeBuildAdminAccess ,AWSCodeBuildDeveloperAccess , et AWSCodeBuildReadOnlyAccess — Mise à jour des politiques existantes</p> | <p>CodeBuild a mis à jour une ressource relative à ces politiques.</p> <p>Les AWSCodeBuildReadOnlyAccess politiques AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess , et ont été modifiées pour mettre à jour une ressource existante. La ressource</p>                                                                                                                                                                                                                         | <p>15 novembre 2024</p>  |

| Modifier                                                                                                                                       | Description                                                                                                                                                                                                                                                                                                                                    | Date          |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
|                                                                                                                                                | d'origine arn:aws:c<br>odebuild:* a été mise<br>à jour versarn:aws:c<br>odebuild:*:*:proje<br>ct/* .                                                                                                                                                                                                                                           |               |
| AWSCodeBuildAdminA<br>ccess ,AWSCodeBu<br>ildDeveloperAccess ,<br>et AWSCodeBuildReadOn<br>lyAccess — Mise à jour<br>des politiques existantes | CodeBuild a ajouté une<br>ressource à ces politiques<br>pour soutenir le changemen<br>t de AWS CodeConnections<br>marque.<br><br>Les AWSCodeBuildReadOn<br>lyAccess politiques<br>AWSCodeBuildAdminA<br>ccess AWSCodeBu<br>ildDeveloperAccess<br>,, et ont été modifiées<br>pour ajouter une ressource<br>,arn:aws:codeconnec<br>tions:*:*:* . | 18 avril 2024 |

| Modifier                                                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                     | Date        |
|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>AWSCodeBuildAdminAccess</b> et <b>AWSCodeBuildDeveloperAccess</b> — Mise à jour des politiques existantes | <p>CodeBuild a ajouté une autorisation à ces politiques pour prendre en charge un type de notification supplémentaire utilisant Amazon Q Developer dans les applications de chat.</p> <p>Les <b>AWSCodeBuildDeveloperAccess</b> politiques <b>AWSCodeBuildAdminAccess</b> et ont été modifiées pour ajouter une autorisation, <code>chatbot:ListMicrosoftTeamsChannelConfigurations</code>.</p> | 16 mai 2023 |
| CodeBuild a commencé à suivre les modifications                                                              | CodeBuild a commencé à suivre les modifications apportées AWS à ses politiques gérées.                                                                                                                                                                                                                                                                                                          | 16 mai 2021 |

## Exemples de politiques gérées par le client

Dans cette section, vous trouverez des exemples de stratégies utilisateur qui accordent des autorisations pour des actions AWS CodeBuild. Ces politiques fonctionnent lorsque vous utilisez l'API CodeBuild AWS SDKs, ou AWS CLI. Lorsque vous utilisez la console, vous devez accorder des autorisations supplémentaires spécifiques à la console. Pour plus d'informations, consultez [Autorisations requises pour utiliser la AWS CodeBuild console](#).

Vous pouvez utiliser les exemples de politiques IAM suivants pour limiter l'accès de vos utilisateurs et de vos rôles.

### Rubriques

- [Autorisation d'un utilisateur à obtenir des informations sur les projets de génération](#)

- [Autoriser un utilisateur à obtenir des informations sur les flottes](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les rapports](#)
- [Autorisation d'un utilisateur à créer des projets de génération](#)
- [Autoriser un utilisateur à créer une flotte](#)
- [Autorisation d'un utilisateur à créer un groupe de rapports](#)
- [Autoriser un utilisateur à supprimer une flotte](#)
- [Autorisation d'un utilisateur à supprimer un groupe de rapports](#)
- [Autorisation d'un utilisateur à supprimer un rapport](#)
- [Autorisation d'un utilisateur à supprimer des projets de génération](#)
- [Autorisation d'un utilisateur à obtenir une liste de noms de projet de génération](#)
- [Autorisation d'un utilisateur à modifier des informations sur les projets de génération](#)
- [Autoriser un utilisateur à modifier une flotte](#)
- [Autorisation d'un utilisateur à modifier un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les générations](#)
- [Autoriser un utilisateur à obtenir une liste des versions IDs pour un projet de construction](#)
- [Autoriser un utilisateur à obtenir une liste de versions IDs](#)
- [Autoriser un utilisateur à obtenir une liste de flottes](#)
- [Autorisation d'un utilisateur à obtenir une liste de groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport](#)
- [Autorisation d'un utilisateur à démarrer l'exécution de générations](#)
- [Autorisation d'un utilisateur à tenter d'arrêter des générations](#)
- [Autorisation d'un utilisateur à tenter de supprimer des générations](#)
- [Permettre à un utilisateur d'obtenir des informations sur les images Docker gérées par CodeBuild](#)
- [Autoriser un utilisateur à ajouter une politique d'autorisation pour un rôle de service de flotte](#)
- [Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC](#)
- [Utilisez une déclaration de refus pour éviter AWS CodeBuild de vous déconnecter des fournisseurs de sources](#)

## Autorisation d'un utilisateur à obtenir des informations sur les projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur des projets de génération dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetProjects",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

## Autoriser un utilisateur à obtenir des informations sur les flottes

L'exemple de déclaration de politique suivant permet à un utilisateur d'obtenir des informations sur les flottes de la us-east-2 région pour créer un compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetFleets",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
 }
]
}
```

## Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir des informations sur les groupes de rapports dans la région us-east-2 du compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetReportGroups",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

## Autorisation d'un utilisateur à obtenir des informations sur les rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir des informations sur les rapports dans la région us-east-2 pour le compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetReports",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

## Autorisation d'un utilisateur à créer des projets de génération

L'exemple de déclaration de politique suivant permet à un utilisateur de créer des projets de construction avec n'importe quel nom, mais uniquement dans la us-east-2 région pour le compte 123456789012 et en utilisant uniquement le rôle de CodeBuild service spécifié :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateProject",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
 }
]
}
```

L'exemple de déclaration de politique suivant permet à un utilisateur de créer des projets de construction avec n'importe quel nom, mais uniquement dans la us-east-2 région pour le compte 123456789012 et uniquement en utilisant le rôle de CodeBuild service spécifié. Cela garantit également que l'utilisateur ne peut utiliser le rôle de service spécifié qu'avec d'autres services AWS CodeBuild et non avec d'autres AWS services.

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateProject",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
 }
]
}
```

```
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole",
 "Condition": {
 "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
 }
 }
]
}
```

### Autoriser un utilisateur à créer une flotte

L'exemple de déclaration de politique suivant permet à un utilisateur de créer une flotte dans la us-east-2 région pour créer un compte 123456789012 :

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateFleet",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
 }
]
}
```

### Autorisation d'un utilisateur à créer un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de créer un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

#### JSON

```
{
 "Version": "2012-10-17",
```



```
"Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

### Autoriser un utilisateur à supprimer une flotte

L'exemple de déclaration de politique suivant permet à un utilisateur de supprimer une flotte dans la us-east-2 région pour créer un compte 123456789012 :

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DeleteFleet",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
 }
]
}
```

### Autorisation d'un utilisateur à supprimer un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de supprimer un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
```

```
"Action": "codebuild:DeleteReportGroup",
"Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
}
]
```

### Autorisation d'un utilisateur à supprimer un rapport

L'exemple suivant d'instruction de stratégie permet à un utilisateur de supprimer un rapport dans la région us-east-2 pour le compte 123456789012 :

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DeleteReport",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

### Autorisation d'un utilisateur à supprimer des projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à supprimer des projets de génération dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

#### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DeleteProject",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

```
}
]
}
```

Autorisation d'un utilisateur à obtenir une liste de noms de projet de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir une liste de noms de projet de génération pour le même compte :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListProjects",
 "Resource": "*"
 }
]
}
```

Autorisation d'un utilisateur à modifier des informations sur les projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à modifier les informations sur des projets de génération avec n'importe quel nom, mais uniquement dans la région us-east-2 pour le compte 123456789012, et à utiliser uniquement le rôle de service AWS CodeBuild spécifié :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:UpdateProject",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
 },
 {
```

```
"Effect": "Allow",
"Action": "iam:PassRole",
"Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
}
]
```

## Autoriser un utilisateur à modifier une flotte

L'exemple de déclaration de politique suivant permet à un utilisateur de modifier une flotte dans la us-east-2 région pour créer un compte 123456789012 :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:UpdateFleet",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
 }
]
}
```

## Autorisation d'un utilisateur à modifier un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de modifier un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:UpdateReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

```
 }
]
}
```

## Autorisation d'un utilisateur à obtenir des informations sur les générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur les générations dans la région us-east-2 pour le compte 123456789012 pour les projets de génération nommés my-build-project et my-other-build-project :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetBuilds",
 "Resource": [
 "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
 "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
]
 }
]
}
```

## Autoriser un utilisateur à obtenir une liste des versions IDs pour un projet de construction

L'exemple de déclaration de politique suivant permet à un utilisateur d'obtenir une liste des constructions IDs dans la us-east-2 région 123456789012 pour tenir compte des projets de construction nommés my-build-project et my-other-build-project :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
```

```
"Action": "codebuild:ListBuildsForProject",
"Resource": [
 "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
 "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
]
}
]
```

Autoriser un utilisateur à obtenir une liste de versions IDs

L'exemple de déclaration de politique suivant permet à un utilisateur d'obtenir une liste de toutes les IDs versions pour le même compte :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListBuilds",
 "Resource": "*"
 }
]
}
```

Autoriser un utilisateur à obtenir une liste de flottes

L'exemple de déclaration de politique suivant permet à un utilisateur d'obtenir une liste des flottes de la us-east-2 région pour créer un compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
```

```
 "Action": "codebuild:ListFleets",
 "Resource": "*"
 }
]
```

Autorisation d'un utilisateur à obtenir une liste de groupes de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de groupes de rapports dans la région us-east-2 pour le compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListReportGroups",
 "Resource": "*"
 }
]
}
```

Autorisation d'un utilisateur à obtenir une liste de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de rapports dans la région us-east-2 pour le compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListReports",
 "Resource": "*"
 }
]
}
```

```
]
}
```

Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de rapports pour un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListReportsForReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```

Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de cas de test pour un rapport dans la région us-east-2 pour le compte 123456789012 :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DescribeTestCases",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
 }
]
}
```



## Autorisation d'un utilisateur à démarrer l'exécution de générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à exécuter des générations dans la région us-east-2 pour le compte 123456789012 pour les projets de génération dont le nom commence par my :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:StartBuild",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

## Autorisation d'un utilisateur à tenter d'arrêter des générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à tenter d'arrêter des générations uniquement dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

### JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:StopBuild",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

## Autorisation d'un utilisateur à tenter de supprimer des générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à tenter de supprimer des générations uniquement dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchDeleteBuilds",
 "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
 }
]
}
```

Permettre à un utilisateur d'obtenir des informations sur les images Docker gérées par CodeBuild

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur toutes les images Docker gérées par CodeBuild :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListCuratedEnvironmentImages",
 "Resource": "*"
 }
]
}
```

## Autoriser un utilisateur à ajouter une politique d'autorisation pour un rôle de service de flotte

L'exemple de déclaration de politique de ressources suivant permet à un utilisateur d'ajouter une politique d'autorisation VPC pour un rôle de service de flotte :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildFleetVpcCreateNI",
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterface"
],
 "Resource": [
 "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
 "arn:aws:ec2:us-west-2:111122223333:security-group/security-
group-id-1",
 "arn:aws:ec2:us-west-2:111122223333:network-interface/*"
]
 },
 {
 "Sid": "CodeBuildFleetVpcPermission",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeDhcpOptions",
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeSubnets",
 "ec2:DescribeVpcs",
 "ec2:ModifyNetworkInterfaceAttribute",
 "ec2>DeleteNetworkInterface"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeBuildFleetVpcNIPermission",
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterfacePermission"
],
 }
]
}
```

```

 "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
 "Condition": {
 "StringEquals": {
 "ec2:Subnet": [
 "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1"
]
 }
 }
]
}

```

L'exemple de déclaration de politique de ressources suivant permet à un utilisateur d'ajouter une politique d'autorisation Amazon Managed Image (AMI) personnalisée pour un rôle de service de flotte :

JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "ec2:DescribeImages",
 "Resource": "*"
 }
]
}

```

L'exemple de déclaration de politique de confiance suivant permet à un utilisateur d'ajouter une politique d'autorisation pour un rôle de service de flotte :

JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildFleetVPCTrustPolicy",

```

```
"Effect": "Allow",
"Principal": {
 "Service": "codebuild.amazonaws.com"
},
"Action": "sts:AssumeRole",
"Condition": {
 "StringEquals": {
 "aws:SourceAccount": "111122223333"
 }
}
}
```

Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC

L'exemple de déclaration de politique suivant accorde AWS CodeBuild l'autorisation de créer une interface réseau dans un VPC avec deux sous-réseaux :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterface",
 "ec2:DescribeDhcpOptions",
 "ec2:DescribeNetworkInterfaces",
 "ec2>DeleteNetworkInterface",
 "ec2:DescribeSubnets",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeVpcs"
],
 "Resource": "*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterfacePermission"
],

```

```

 "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
 "Condition": {
 "StringEquals": {
 "ec2:AuthorizedService": "codebuild.amazonaws.com"
 },
 "ArnEquals": {
 "ec2:Subnet": [
 "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
 "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-2"
]
 }
 }
 }
]
}

```

Utilisez une déclaration de refus pour éviter AWS CodeBuild de vous déconnecter des fournisseurs de sources

L'exemple de déclaration de stratégie suivant utilise une déclaration de refus pour empêcher AWS CodeBuild de se déconnecter des fournisseurs de source. Il utilise `codebuild:DeleteAuthToken`, qui est l'inverse de `codebuild:PersistAuthToken` et de `codebuild:ImportSourceCredentials`, pour se connecter aux fournisseurs de source. Pour de plus amples informations, veuillez consulter [Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources](#).

## JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": "codebuild:DeleteAuthToken",
 "Resource": "*"
 }
]
}

```

## AWS CodeBuild référence aux autorisations

Vous pouvez utiliser des AWS clés de condition larges dans vos AWS CodeBuild polices pour exprimer des conditions. Pour obtenir une liste, consultez la section [Clés disponibles](#) dans le guide de l'utilisateur IAM.

Vous spécifiez les actions dans le champ `Action` de la politique. Pour spécifier une action, utilisez le préfixe `codebuild:` suivi du nom de l'opération d'API (par exemple, `codebuild:CreateProject` ou `codebuild:StartBuild`). Pour spécifier plusieurs actions dans une même instruction, séparez-les par une virgule (par exemple, `"Action": [ "codebuild:CreateProject", "codebuild:StartBuild" ]`).

### Utilisation de caractères génériques

Vous spécifiez un ARN, avec ou sans caractère générique (\*) comme valeur de ressource dans le champ `Resource` de la stratégie. Vous pouvez utiliser un caractère générique pour spécifier plusieurs actions ou ressources. Par exemple, `codebuild:*` spécifie toutes les CodeBuild actions et `codebuild:Batch*` indique toutes les CodeBuild actions commençant par le mot `Batch`. L'exemple suivant accorde l'accès à tous les projets de génération dont le nom commence par `my` :

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

### CodeBuild Opérations d'API et autorisations requises pour les actions

#### BatchDeleteBuilds

Action : `codebuild:BatchDeleteBuilds`

Requête pour supprimer des générations

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetBuilds

Action : `codebuild:BatchGetBuilds`

Requête pour obtenir des informations sur des générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetProjects

Action : `codebuild:BatchGetProjects`

Requise pour obtenir des informations sur des projets de génération.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetReportGroups

Action : `codebuild:BatchGetReportGroups`

Obligatoire pour obtenir des informations sur les groupes de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### BatchGetReports

Action : `codebuild:BatchGetReports`

Obligatoire pour obtenir des informations sur des générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### BatchPutTestCases<sup>1</sup>

Action : `codebuild:BatchPutTestCases`

Obligatoire pour créer ou mettre à jour un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### CreateProject

Actions : `codebuild>CreateProject`, `iam:PassRole`

Requise pour créer des projets de génération.

Ressources:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

#### CreateReport<sup>1</sup>

Action : `codebuild>CreateReport`



Obligatoire pour créer un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### CreateReportGroup

Action : `codebuild:CreateReportGroup`

Obligatoire pour créer un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### CreateWebhook

Action : `codebuild:CreateWebhook`

Requise pour créer un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### DeleteProject

Action : `codebuild>DeleteProject`

Nécessaire pour supprimer un CodeBuild projet.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### DeleteReport

Action : `codebuild>DeleteReport`

Obligatoire pour supprimer une règle.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### DeleteReportGroup

Action : `codebuild>DeleteReportGroup`

Obligatoire pour supprimer un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## DeleteSourceCredentials

Action : `codebuild>DeleteSourceCredentials`

Nécessaire pour supprimer un ensemble d'`SourceCredentialsInfo`objets contenant des informations sur les informations d'identification d'un GitHub référentiel GitHub Enterprise Server ou Bitbucket.

Ressource : \*

## DeleteWebhook

Action : `codebuild>DeleteWebhook`

Requête pour créer un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## DescribeTestCases

Action : `codebuild>DescribeTestCases`

Obligatoire pour renvoyer une liste paginée de cas de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## ImportSourceCredentials

Action : `codebuild>ImportSourceCredentials`

Nécessaire pour importer un ensemble d'`SourceCredentialsInfo`objets contenant des informations sur les informations d'identification d'un GitHub référentiel GitHub Enterprise Server ou Bitbucket.

Ressource : \*

## InvalidateProjectCache

Action : `codebuild>InvalidateProjectCache`

Requête pour réinitialiser le cache pour un projet.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### ListBuildBatches

Action : `codebuild:ListBuildBatches`

Nécessaire pour obtenir une liste des lots de construction IDs.

Ressource : \*

#### ListBuildBatchesForProject

Action : `codebuild:ListBuildBatchesForProject`

Nécessaire pour obtenir une liste des lots de construction IDs pour un projet spécifique.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### ListBuilds

Action : `codebuild:ListBuilds`

Nécessaire pour obtenir une liste des versions IDs.

Ressource : \*

#### ListBuildsForProject

Action : `codebuild:ListBuildsForProject`

Nécessaire pour obtenir la liste des versions IDs d'un projet de construction.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### ListCuratedEnvironmentImages

Action : `codebuild:ListCuratedEnvironmentImages`

Requise pour obtenir des informations sur toutes les images Docker gérées par AWS CodeBuild.

Ressource : \* (requise, mais ne fait pas référence à une ressource AWS adressable)

#### ListProjects

Action : `codebuild:ListProjects`

Requise pour obtenir une liste de noms de projet de génération.

Ressource : \*

### ListReportGroups

Action : `codebuild:ListReportGroups`

Obligatoire pour obtenir une liste des groupes de rapports.

Ressource : \*

### ListReports

Action : `codebuild:ListReports`

Obligatoire pour obtenir une liste de rapports.

Ressource : \*

### ListReportsForReportGroup

Action : `codebuild:ListReportsForReportGroup`

Obligatoire pour obtenir une liste de rapports pour un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### RetryBuild

Action : `codebuild:RetryBuild`

Nécessaire pour réessayer les builds.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### StartBuild

Action : `codebuild:StartBuild`

Requise pour démarrer l'exécution de générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## StopBuild

Action : `codebuild:StopBuild`

Requise pour tenter d'arrêter des générations en cours d'exécution.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## UpdateProject

Actions : `codebuild:UpdateProject`, `iam:PassRole`

Requise pour modifier des informations sur les générations.

Ressources:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

## UpdateProjectVisibility

Actions : `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Nécessaire pour modifier la visibilité publique des versions d'un projet.

Ressources:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

## UpdateReport<sup>1</sup>

Action : `codebuild:UpdateReport`

Obligatoire pour créer ou mettre à jour un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## UpdateReportGroup

Action : `codebuild:UpdateReportGroup`

Obligatoire pour mettre à jour un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## UpdateWebhook

Action : `codebuild:UpdateWebhook`

Requise pour mettre à jour un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

<sup>1</sup> Utilisé uniquement à des fins d'autorisation. Il n'existe aucune API pour cette action.

## Utilisation de balises pour contrôler l'accès aux AWS CodeBuild ressources

Les conditions figurant dans les déclarations de politique IAM font partie de la syntaxe que vous pouvez utiliser pour spécifier les autorisations relatives aux actions basées sur des CodeBuild projets. Vous pouvez créer une politique qui autorise ou refuse des actions sur des projets en fonction des balises associées à ces projets, puis appliquer ces politiques aux groupes IAM que vous configurez pour gérer les utilisateurs. Pour plus d'informations sur l'application de balises à un projet à l'aide de la console ou AWS CLI consultez [Créez un projet de construction dans AWS CodeBuild](#). Pour plus d'informations sur l'application de balises à l'aide du CodeBuild SDK, consultez [CreateProject](#) la section « [Tags](#) » dans le manuel de référence des CodeBuild API. Pour plus d'informations sur l'utilisation de balises pour contrôler l'accès aux AWS ressources, consultez la section [Contrôle de l'accès aux AWS ressources à l'aide de balises de ressources](#) dans le guide de l'utilisateur IAM.

### Important

Lorsque vous utilisez la fonctionnalité de capacité réservée, les données mises en cache sur les instances de flotte, y compris les fichiers source, les couches Docker et les répertoires mis en cache spécifiés dans les spécifications de construction, peuvent être accessibles aux autres projets du même compte. Cela est intentionnel et permet aux projets au sein d'un même compte de partager des instances de flotte.

### Exemple Exemple 1 : Limiter les actions CodeBuild du projet en fonction des balises de ressources

L'exemple suivant refuse toutes les actions `BatchGetProjects` sur les projets marqués avec la clé `Environment` utilisant la valeur `Production`. L'administrateur d'un utilisateur doit associer

cette politique IAM en plus de la politique utilisateur gérée aux utilisateurs non autorisés. La clé de condition `aws:ResourceTag` est utilisée pour contrôler l'accès aux ressources en fonction de leurs balises.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:BatchGetProjects"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:ResourceTag/Environment": "Production"
 }
 }
 }
]
}
```

Exemple Exemple 2 : Limiter les actions CodeBuild du projet en fonction des balises de demande

La stratégie suivante refuse aux utilisateurs l'autorisation pour l'action `CreateProject` si la demande contient une balise avec la clé `Environment` et la valeur de clé `Production`. En outre, la stratégie empêche ces utilisateurs non autorisés de modifier des projets en utilisant la clé de condition `aws:TagKeys` pour ne pas autoriser `UpdateProject` si la requête contient une balise avec la clé `Environment`. Un administrateur doit associer cette politique IAM en plus de la politique utilisateur gérée aux utilisateurs qui ne sont pas autorisés à effectuer ces actions. La clé de `aws:RequestTag` condition est utilisée pour contrôler les balises qui peuvent être transmises dans une demande IAM

JSON

```
{
 "Version": "2012-10-17",
```

```
"Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:CreateProject"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:RequestTag/Environment": "Production"
 }
 }
 },
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:UpdateProject"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:TagKeys": ["Environment"]
 }
 }
 }
]
```

Exemple Exemple 3 : Refuser ou autoriser des actions sur des groupes de rapports en fonction des balises de ressources

Vous pouvez créer une politique qui autorise ou refuse les actions sur les CodeBuild ressources (projets et groupes de rapports) en fonction des AWS balises associées à ces ressources, puis appliquer ces politiques aux groupes IAM que vous configurez pour gérer les utilisateurs. Par exemple, vous pouvez créer une politique qui refuse toutes les CodeBuild actions sur tout groupe de rapports dont la clé de AWS balise Status et la valeur clé sont Secret, puis appliquer cette politique au groupe IAM que vous avez créé pour les développeurs généraux (*Developers*). Vous devez ensuite vous assurer que les développeurs travaillant sur ces groupes de rapports balisés ne sont pas membres de ce *Developers* groupe général, mais appartiennent plutôt à un autre groupe IAM auquel la politique restrictive n'est pas appliquée (SecretDevelopers).



L'exemple suivant refuse toutes les CodeBuild actions sur les groupes de rapports marqués par la clé Status et la valeur clé de Secret :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:BatchGetReportGroups",
 "codebuild:CreateReportGroup",
 "codebuild>DeleteReportGroup",
 "codebuild:ListReportGroups",
 "codebuild:ListReportsForReportGroup",
 "codebuild:UpdateReportGroup"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "aws:RequestedRegion": "us-east-1"
 }
 }
 }
]
}
```

Exemple Exemple 4 : Limiter CodeBuild les actions en AWSCode BuildDeveloperAccess fonction des balises de ressources

Vous pouvez créer des politiques qui autorisent CodeBuild des actions sur tous les groupes de rapports et projets qui ne sont pas marqués par des balises spécifiques. Par exemple, la stratégie suivante accorde l'équivalent des autorisations [AWSCodeBuildDeveloperAccess](#) pour tous les groupes de rapports, sauf ceux balisés avec les balises spécifiées :

JSON

```
{
 "Version": "2012-10-17",
```

```

"Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codebuild:StartBuild",
 "codebuild:StopBuild",
 "codebuild:BatchGet*",
 "codebuild:GetResourcePolicy",
 "codebuild:DescribeTestCases",
 "codebuild:List*",
 "codecommit:GetBranch",
 "codecommit:GetCommit",
 "codecommit:GetRepository",
 "codecommit:ListBranches",
 "cloudwatch:GetMetricStatistics",
 "events:DescribeRule",
 "events:ListTargetsByRule",
 "events:ListRuleNamesByTarget",
 "logs:GetLogEvents",
 "s3:GetBucketLocation",
 "s3:ListAllMyBuckets"
],
 "Resource": "*",
 "Condition": {
 "StringNotEquals": {
 "aws:ResourceTag/Status": "Secret",
 "aws:ResourceTag/Team": "Saanvi"
 }
 }
 }
]
}

```

## Affichage des ressources dans la console

La AWS CodeBuild console doit être ListRepositories autorisée à afficher la liste des référentiels de votre AWS compte dans la AWS région où vous êtes connecté. La console comprend également une fonction Go to resource (Accéder aux ressources) qui permet d'effectuer rapidement une recherche de ressources sensible à la casse. Cette recherche est effectuée dans votre AWS compte dans la AWS région où vous êtes connecté. Les ressources suivantes sont affichées pour les services suivants :

- AWS CodeBuild : Projets de génération
- AWS CodeCommit : Référentiels
- AWS CodeDeploy : Applications
- AWS CodePipeline : Pipelines

Pour effectuer cette recherche pour les ressources dans tous les services, vous devez disposer des autorisations suivantes :

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Les résultats ne sont pas renvoyés pour les ressources d'un service si vous ne disposez pas d'autorisations pour ce service. Même si vous êtes autorisé à afficher des ressources, certaines ressources ne sont pas renvoyées si une valeur Deny explicite est définie pour l'affichage de ces ressources.

## Validation de conformité pour AWS CodeBuild

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

# Résilience dans AWS CodeBuild

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

## Sécurité de l'infrastructure dans AWS CodeBuild

En tant que service géré, AWS CodeBuild il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder CodeBuild via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

## Accédez à votre fournisseur source dans CodeBuild

Pour GitHub ou GitHub Enterprise Server, vous utilisez un jeton d'accès personnel, un secret du Gestionnaire de Secrets, une connexion ou une OAuth application pour accéder au fournisseur source. Pour Bitbucket, vous utilisez un jeton d'accès, un mot de passe d'application, un secret du Gestionnaire de Secrets, une connexion ou une OAuth application pour accéder au fournisseur source.

### Rubriques

- [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#)
- [GitHub et accès au serveur d' GitHub entreprise dans CodeBuild](#)
- [Accès à Bitbucket dans CodeBuild](#)
- [GitLab accéder à CodeBuild](#)

## Créez et stockez un jeton dans un secret du Gestionnaire de Secrets

Si vous choisissez de stocker votre jeton d'accès à l'aide de Secrets Manager, vous pouvez utiliser une connexion secrète existante ou en créer un nouveau. Pour créer un nouveau secret, procédez comme suit :

### AWS Management Console

Pour créer un secret Secrets Manager dans le AWS Management Console

1. Dans le champ Source provider, choisissez Bitbucket ou GitHub Enterprise. GitHub
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez Informations d'identification source par défaut pour utiliser les informations d'identification source par défaut de votre compte pour les appliquer à tous les projets.
    - a. Si vous n'êtes pas connecté à votre fournisseur de source, choisissez Gérer les informations d'identification de source par défaut.
    - b. Pour le type d'identification, choisissez un type d'identification autre que CodeConnections
    - c. Pour Service, choisissez Secrets Manager et pour Secrets, choisissez Nouveau secret.
    - d. Dans Nom du secret, entrez le nom de votre secret.
    - e. Dans Description secrète - facultatif, entrez une description de votre secret.
    - f. Selon le fournisseur source que vous avez choisi, entrez votre jeton ou votre nom d'utilisateur et le mot de passe de l'application, puis choisissez Enregistrer.
  - Choisissez Identifiant source personnalisé pour utiliser un identifiant source personnalisé afin de remplacer les paramètres par défaut de votre compte.
    - a. Pour le type d'identification, choisissez un type d'identification autre que CodeConnections
    - b. Dans Connexion, choisissez Créer un secret.

- c. Dans Nom du secret, entrez le nom de votre secret.
- d. Dans Description secrète - facultatif, entrez une description de votre secret.
- e. Selon le fournisseur source que vous avez choisi, entrez votre jeton ou votre nom d'utilisateur et le mot de passe de l'application, puis choisissez Créer.

## AWS CLI

Pour créer un secret Secrets Manager dans le AWS CLI

- Ouvrez une invite de terminal (Linux, macOS ou Unix) ou de commande (Windows). Utilisez le AWS CLI pour exécuter la create-secret commande Secrets Manager.

```
aws secretsmanager create-secret --region <aws-region> \
 --name '<secret-name>' \
 --description '<secret-description>' \
 --secret-string '{
 "ServerType": "<server-type>",
 "AuthType": "<auth-type>",
 "Token": "<token>"
 }' \
 --tags Key=codebuild:source,Value='' \
 Key=codebuild:source:type,Value=<type> \
 Key=codebuild:source:provider,Value=<provider>
```

Les secrets de Secrets Manager qui sont CodeBuild acceptés doivent se trouver dans le même compte et dans la même AWS région que le CodeBuild projet et doivent être au format JSON suivant :

```
{
 "ServerType": ServerType,
 "AuthType": AuthType,
 "Token": string,
 "Username": string // Optional and is only used for Bitbucket app
 password
}
```

| Champ             | Valeurs valides                          | Description                                                                                                                                                                                         |
|-------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ServerType        | GITHUB<br>GITHUB_ENTREPRISE<br>BITBUCKET | Le fournisseur de source tiers pour votre secret Secrets Manager.                                                                                                                                   |
| AuthType          | JETON D'ACCÈS PERSONNEL<br>BASIC_AUTH    | Type de jeton d'accès utilisé par les informations d'identification. Car GitHub seul PERSONAL_ACCESS_TOKEN est valide. BASIC_AUTH n'est valide que pour le mot de passe de l'application Bitbucket. |
| Jeton             | <i>string</i>                            | Pour GitHub notre GitHub Enterprise, il s'agit du jeton d'accès personnel. Pour Bitbucket, il s'agit du jeton d'accès ou du mot de passe de l'application Bitbucket.                                |
| Nom d'utilisateur | <i>string</i>                            | Le nom d'utilisateur Bitbucket lorsqu'il AuthType est BASIC_AUTH. Ce paramètre n'est pas valide pour les autres types de fournisseurs de sources.                                                   |

En outre, CodeBuild utilise les balises de ressources suivantes sur le secret pour garantir que les secrets sont facilement sélectionnables lors de la création ou de la modification de projets.

| Clé de balise                    | Valeur de balise        | Description                                                        |
|----------------------------------|-------------------------|--------------------------------------------------------------------|
| codebuild : source : fournisseur | github                  | CodeBuild Indique à quel fournisseur ce secret est destiné.        |
|                                  | github_entreprise       |                                                                    |
|                                  | bitbucket               |                                                                    |
| codebuild : source : type        | jeton d'accès personnel | CodeBuild Indique le type de jeton d'accès contenu dans ce secret. |
|                                  | basic_auth              |                                                                    |

## GitHub et accès au serveur d' GitHub entreprise dans CodeBuild

En GitHub effet, vous pouvez utiliser un jeton d'accès personnel, une OAuth application, un secret du Gestionnaire de Secrets ou une connexion à une GitHub application pour accéder au fournisseur source. Pour GitHub Enterprise Server, vous pouvez utiliser un jeton d'accès personnel, un secret du Gestionnaire de Secrets ou une connexion GitHub App pour accéder au fournisseur source.

### Rubriques

- [GitHub Connexions aux applications pour GitHub et GitHub Enterprise Server](#)
- [GitHub et jeton d'accès au serveur GitHub Enterprise](#)
- [GitHub OAuth appli](#)

## GitHub Connexions aux applications pour GitHub et GitHub Enterprise Server

Vous pouvez utiliser GitHub l'application pour vous connecter à CodeBuild. GitHub Les connexions aux applications sont prises en charge via [AWS CodeConnections](#).

L'accès au fournisseur de source vous permet de déclencher une compilation en vous abonnant à [GitHub événements webhook](#) [CreateWebhook](#)using ou [Tutoriel : Configuration d'un CodeBuild exécutateur d' GitHub actions hébergé](#) en utilisant in CodeBuild.



 Note

CodeConnections est disponible dans un nombre de régions inférieur à CodeBuild. Vous pouvez utiliser des connexions entre régions dans CodeBuild. Les connexions créées dans les régions optionnelles ne peuvent pas être utilisées dans d'autres régions. Pour plus d'informations, consultez [Points de terminaison et quotas AWS CodeConnections](#).

## Rubriques

- [Étape 1 : créer une connexion à GitHub l'application \(console\)](#)
- [Étape 2 : Accorder l'accès au rôle IAM du CodeBuild projet pour utiliser la connexion](#)
- [Étape 3 : Configuration CodeBuild pour utiliser la nouvelle connexion](#)
- [Résolution des problèmes liés à l' GitHub application](#)

## Étape 1 : créer une connexion à GitHub l'application (console)

Suivez ces étapes pour utiliser la CodeBuild console afin d'ajouter une connexion pour votre projet dans GitHub.

Pour créer une connexion avec GitHub

- Suivez les instructions du guide de l'utilisateur des outils de développement pour [créer une connexion à GitHub](#).

 Note

Au lieu de créer ou d'utiliser une connexion existante dans votre compte, vous pouvez utiliser une connexion partagée depuis un autre AWS compte. Pour plus d'informations, voir [Partager des connexions avec AWS des comptes](#).

## Étape 2 : Accorder l'accès au rôle IAM du CodeBuild projet pour utiliser la connexion

Vous pouvez accorder l'accès au rôle IAM du CodeBuild projet pour utiliser les GitHub jetons vendus par votre connexion.

Pour accorder l'accès CodeBuild au rôle IAM du projet

1. Créez un rôle IAM pour votre CodeBuild projet en suivant les instructions correspondant [CodeBuild Autoriser l'interaction avec d'autres AWS services](#) à votre CodeBuild projet.
2. Tout en suivant les instructions, ajoutez la politique IAM suivante à votre rôle de CodeBuild projet pour autoriser l'accès à la connexion.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codeconnections:GetConnectionToken",
 "codeconnections:GetConnection"
],
 "Resource": [
 "arn:aws:iam::*:role/Service*"
]
 }
]
}
```

### Étape 3 : Configuration CodeBuild pour utiliser la nouvelle connexion

Vous pouvez configurer une connexion en tant qu'identifiant au niveau du compte et l'utiliser dans un projet.

#### AWS Management Console

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS Management Console

1. Dans le champ Source provider, sélectionnez GitHub.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez Informations d'identification source par défaut pour utiliser les informations d'identification source par défaut de votre compte pour les appliquer à tous les projets.

- a. Si vous n'êtes pas connecté à GitHub, choisissez Gérer les informations d'identification de source par défaut.
- b. Pour le type d'identifiant, choisissez GitHub App.
- c. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.
- Choisissez Identifiant source personnalisé pour utiliser un identifiant source personnalisé afin de remplacer les paramètres par défaut de votre compte.
  - a. Pour le type d'identifiant, choisissez GitHub App.
  - b. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.

## AWS CLI

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS CLI

- Ouvrez une invite de terminal (Linux, macOS ou Unix) ou de commande (Windows). Utilisez le AWS CLI pour exécuter la `import-source-credentials` commande, en spécifiant le `--auth-type` `--server-type`, et `--token` pour votre connexion.

Utilisez la commande suivante :

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type GITHUB --token <connection-arn>
```

Vous pouvez également configurer plusieurs jetons pour vos CodeBuild projets. Pour de plus amples informations, veuillez consulter [Configurer plusieurs jetons comme informations d'identification au niveau de la source](#).

## Résolution des problèmes liés à l' GitHub application

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés à l' GitHub application.

### Rubriques

- [Installez l' GitHub application AWS Connector for dans une région indésirable](#)

- [La connexion à GitHub l'application n'a pas accès aux référentiels](#)
- [Le rôle IAM du AWS service ne dispose pas des autorisations IAM nécessaires.](#)

Installez l' GitHub application AWS Connector for dans une région indésirable

Problème : vous avez installé le AWS connecteur GitHub depuis le GitHub Marketplace, mais la connexion a été créée dans une région indésirable. Si vous tentez de reconfigurer l'application sur le GitHub site Web, elle ne fonctionnera pas car elle est déjà installée sur votre GitHub compte.

Cause possible : l'application étant déjà installée sur votre GitHub compte, vous ne pouvez que reconfigurer les autorisations de l'application.

Solution recommandée : vous pouvez créer une nouvelle connexion avec l'ID d'installation dans la région souhaitée.

1. Ouvrez la CodeConnections console dans <https://console.aws.amazon.com/codesuite/Paramètres/connexions> et naviguez jusqu'à la région souhaitée à l'aide du sélecteur de région dans la barre de navigation de la AWS console.
2. Suivez les instructions du guide de l'utilisateur des outils de développement pour [créer une connexion à GitHub](#).

 Note

Comme vous avez déjà installé le AWS Connector pour GitHub l'application, vous pouvez le choisir au lieu d'installer une nouvelle application.

La connexion à GitHub l'application n'a pas accès aux référentiels

Problème : un AWS service utilisant la connexion, tel que CodeBuild ou CodePipeline, indique qu'il n'a pas accès au référentiel ou que le référentiel n'existe pas. Voici quelques messages d'erreur possibles :

- Authentication required for primary source.
- Unable to create webhook at this time. Please try again later.
- Failed to create webhook. GitHub API limit reached. Please try again later.

Cause possible : il se peut que vous utilisiez l' GitHub application et que vous n'ayez pas accordé l'autorisation d'accès au webhook.

Solution recommandée : pour accorder l'étendue d'autorisation requise, suivez les instructions de la [section Navigation vers l' GitHub application que vous souhaitez consulter ou modifier](#) pour configurer l'application installée. Dans la section des autorisations, vous verrez que l'application n'a pas d'autorisation pour les webhooks et qu'il est possible de consulter les autorisations récemment demandées. Vérifiez et acceptez les nouvelles autorisations. Pour plus d'informations, consultez la section [Approbation des autorisations mises à jour pour une application](#). GitHub

Cause possible : La connexion fonctionnait comme prévu, mais elle n'a soudainement plus accès aux référentiels.

Solution possible : Commencez par examiner vos [autorisations](#) et vos [installations](#), puis vérifiez que l' GitHub application est autorisée et installée. Si l'installation de l' GitHub application est suspendue, vous devez l'annuler. Si l' GitHub application n'est pas autorisée pour une connexion [UAT \(User Access Token\)](#), ou si elle n'est pas installée pour une connexion [IAT \(Installation Access Token\)](#), la connexion existante n'est plus utilisable et vous devrez créer une nouvelle connexion. Notez que la réinstallation de l' GitHub application ne rétablira pas la connexion précédente associée à l'ancienne installation.

Solution possible : s'il s'agit d'une connexion UAT, assurez-vous qu'elle n'est pas utilisée simultanément, par exemple lors de plusieurs exécutions CodeBuild simultanées de génération. Cela est dû au fait qu'il invalide GitHub immédiatement un UAT précédemment émis si un jeton expirant est actualisé par la connexion. Si vous devez utiliser une connexion UAT pour plusieurs CodeBuild versions simultanées, vous pouvez créer plusieurs connexions et utiliser chaque connexion indépendamment.

Solution possible : Si la connexion UAT n'a pas été utilisée au cours des 6 derniers mois, la connexion sera invalidée par. GitHub Pour résoudre ce problème, créez une nouvelle connexion.

Cause possible : vous avez peut-être utilisé une connexion UAT sans installer l'application.

Solution recommandée : Bien que la création d'une connexion UAT ne nécessite pas d'associer la connexion à une installation d' GitHub application, une installation est requise pour que le référentiel soit accessible. Suivez les instructions pour [vérifier les installations](#) afin de vous assurer que l' GitHub application est bien installée. S'il n'est pas installé, accédez à la [page de GitHub l'application](#) pour installer l'application. Pour plus d'informations sur l'accès de l'UAT, consultez [À propos des jetons d'accès utilisateur](#).

Le rôle IAM du AWS service ne dispose pas des autorisations IAM nécessaires.

Problème : l'un des messages d'erreur suivants s'affiche :

- Access denied to connection `<connection-arn>`
- Failed to get access token from `<connection-arn>`

Solution recommandée : vous utilisez généralement une connexion avec un AWS service, tel que CodePipeline ou CodeBuild. Lorsque vous attribuez AWS un rôle IAM au AWS service, celui-ci peut utiliser l'autorisation du rôle pour agir en votre nom. Assurez-vous que le rôle IAM dispose des autorisations nécessaires. Pour plus d'informations sur l'autorisation IAM nécessaire, voir [Accorder l'accès au rôle IAM CodeBuild du projet pour utiliser la connexion et la gestion des identités et des accès pour les AWS CodeStar notifications et CodeConnections](#) dans le guide de l'utilisateur de la console Developer Tools.

## GitHub et jeton d'accès au serveur GitHub Enterprise

### Prérequis de jeton d'accès

Avant de commencer, vous devez ajouter les étendues d'autorisation appropriées à votre jeton GitHub d'accès.

En effet GitHub, votre jeton d'accès personnel doit avoir les champs d'application suivants.

- `repo` : accorde un contrôle total des référentiels privés.
- `repo:status` : accorde l' read/write accès aux statuts de validation des référentiels publics et privés.
- `admin:repo_hook` : accorde le contrôle total des hooks de référentiel. Cette règle n'est pas obligatoire si votre jeton dispose de la règle `repo`.
- `admin:org_hook` : accorde le contrôle total des hooks de l'organisation. Cette étendue n'est requise que si vous utilisez la fonctionnalité de webhook de l'organisation.

Pour plus d'informations, consultez [la section Comprendre les étendues des OAuth applications](#) sur le GitHub site Web.

Si vous utilisez des jetons d'accès personnels précis, selon votre cas d'utilisation, votre jeton d'accès personnel peut nécessiter les autorisations suivantes :

- `Contenu : Lecture seule` : autorise l'accès aux référentiels privés. Cette autorisation est requise si vous utilisez des référentiels privés comme source.

- Statuts de validation : lecture et écriture : accorde l'autorisation de créer des statuts de validation. Cette autorisation est requise si le webhook est configuré dans votre projet ou si la fonctionnalité d'état de création de rapports est activée.
- Webhooks : lecture et écriture : accorde l'autorisation de gérer les webhooks. Cette autorisation est requise si le webhook est configuré dans votre projet.
- Demandes d'extraction : en lecture seule : accorde l'autorisation d'accéder aux demandes d'extraction. Cette autorisation est requise si votre webhook possède un FILE\_PATH filtre sur les événements de pull request.
- Administration : lecture et écriture : cette autorisation est requise si vous utilisez la fonctionnalité GitHub Actions Runner auto-hébergée avec CodeBuild. Pour plus de détails, voir [Créer un jeton d'enregistrement pour un référentiel](#) et [Tutoriel : Configuration d'un CodeBuild exécutateur d' GitHub actions hébergé](#).

 Note

Si vous souhaitez accéder aux référentiels de l'organisation, assurez-vous de spécifier l'organisation en tant que propriétaire de la ressource du jeton d'accès.

Pour plus d'informations, consultez la section [Autorisations requises pour les jetons d'accès personnels détaillés](#) sur le GitHub site Web.

Connect GitHub avec un jeton d'accès (console)

Pour utiliser la console afin de connecter votre projet à GitHub l'aide d'un jeton d'accès, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Dans le champ Source provider, sélectionnez GitHub.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez d'utiliser les informations d'identification du compte pour appliquer les informations d'identification source par défaut de votre compte à tous les projets.
    - a. Si vous n'êtes pas connecté à GitHub, choisissez Gérer les informations d'identification du compte.
    - b. Pour le type d'identifiant, choisissez jeton d'accès personnel.

- Si vous avez choisi d'utiliser les informations d'identification au niveau du compte pour le service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :
  - a. Si vous choisissez d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou d'en créer un nouveau, puis choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
  - b. Si vous choisissez de l'utiliser CodeBuild, entrez votre jeton d'accès GitHub personnel, puis choisissez Enregistrer.
- Sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement pour utiliser des informations d'identification source personnalisées afin de remplacer les paramètres d'identification de votre compte.
  - a. Dans la liste des informations d'identification renseignée, choisissez l'une des options sous Jeton d'accès personnel.
  - b. Vous pouvez également créer un nouveau jeton d'accès personnel en sélectionnant créer une nouvelle connexion par jeton d'accès personnel dans la description.

Connectez-vous à GitHub l'aide d'un jeton d'accès (CLI)

Suivez ces étapes pour connecter votre projet AWS CLI à l' GitHub aide d'un jeton d'accès. Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

1. Exécutez la commande import-source-credentials :

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, *import-source-credentials.json*) situé sur l'ordinateur local ou sur l'instance où le AWS CLI est installé. Modifiez les données copiées comme suit, puis enregistrez vos résultats.

```
{
 "serverType": "server-type",
 "authType": "auth-type",
```



```
"shouldOverwrite": "should-overwrite",
"token": "token",
"username": "username"
}
```

Remplacez les éléments suivants :

- *server-type*: valeur requise. Le fournisseur de source utilisé pour cette information d'identification. Les valeurs valides sont GITHUB, BITBUCKET, GITHUB\_ENTERPRISE, GITLAB et GITLAB\_SELF\_MANAGED.
  - *auth-type*: valeur requise. Type d'authentification utilisé pour se connecter à un référentiel. Les valeurs valides sont OAUTH, BASIC\_AUTH, PERSONAL\_ACCESS\_TOKEN, CODECONNECTIONS et SECRETS\_MANAGER. Car GitHub seul PERSONAL\_ACCESS\_TOKEN est autorisé. BASIC\_AUTH n'est autorisé qu'avec le mot de passe de l'application Bitbucket.
  - *should-overwrite*: valeur facultative. Définissez cette valeur sur `false` pour empêcher le remplacement des informations d'identification source du référentiel. Définissez cette valeur sur `true` pour remplacer les informations d'identification source du référentiel. La valeur par défaut est `true`.
  - *token*: valeur requise. Pour GitHub notre serveur d' GitHub entreprise, il s'agit du jeton d'accès personnel. Pour Bitbucket, il s'agit du jeton d'accès personnel ou du mot de passe de l'application. Pour le type d'authentification CODECONNECTIONS, il s'agit de l'ARN de connexion. Pour le type d'authentification SECRETS\_MANAGER, il s'agit de l'ARN secret.
  - *username*: valeur facultative. Ce paramètre est ignoré pour GitHub les fournisseurs de sources GitHub Enterprise Server.
2. Pour vous connecter à votre compte avec un jeton d'accès, passez au répertoire contenant le fichier `import-source-credentials.json` que vous avez enregistré à l'étape 1 et exécutez à nouveau la commande `import-source-credentials`.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Les données au format JSON apparaissent dans la sortie avec un Amazon Resource Name (ARN).

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

```
}
```

### Note

Si vous exécutez la commande `import-source-credentials` avec le même type de serveur et de auth une seconde fois, le jeton d'accès stocké est mis à jour.

Une fois que votre compte est connecté par un jeton d'accès, vous pouvez l'utiliser `create-project` pour créer votre CodeBuild projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

3. Pour afficher les jetons d'accès connectés, exécutez la commande `list-source-credentials`.

```
aws codebuild list-source-credentials
```

Un objet `sourceCredentialsInfos` au format JSON apparaît dans la sortie :

```
{
 "sourceCredentialsInfos": [
 {
 "authType": "auth-type",
 "serverType": "server-type",
 "arn": "arn"
 }
]
}
```

Le `sourceCredentialsObject` contient la liste des informations d'identification de la source connectée :

- L'élément `authType` est le type d'authentification utilisé par les informations d'identification. Cela peut être `OAUTHBASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS`, ou `SECRETS_MANAGER`.
- Le `serverType` est le type de fournisseur de source. Cela peut être `GITHUBGITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB`, ou `GITLAB_SELF_MANAGED`.
- Le `arn` est l'ARN du jeton.

4. Pour vous déconnecter d'un fournisseur de source et supprimer ses jetons d'accès, exécutez la commande `delete-source-credentials` avec son ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Des données au format JSON sont renvoyées avec un ARN des informations d'identification supprimées.

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

## GitHub OAuth appli

### Connect GitHub en utilisant OAuth (console)

Pour utiliser la console afin de connecter votre projet à GitHub une OAuth application, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Dans le champ Source provider, sélectionnez GitHub.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez d'utiliser les informations d'identification du compte pour appliquer les informations d'identification source par défaut de votre compte à tous les projets.
    - a. Si vous n'êtes pas connecté à GitHub, choisissez Gérer les informations d'identification du compte.
    - b. Pour le type d'identifiant, choisissez OAuth app.
  - Si vous avez choisi d'utiliser les informations d'identification au niveau du compte pour le service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :
    - a. Si vous choisissez d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou d'en créer un nouveau, puis choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
    - b. Si vous choisissez d'utiliser, CodeBuild puis choisissez Enregistrer.

- Sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement pour utiliser des informations d'identification source personnalisées afin de remplacer les paramètres d'identification de votre compte.
  - a. Dans la liste des informations d'identification renseignée, choisissez l'une des options sous OAuth application.
  - b. Vous pouvez également créer un nouveau jeton d' OAuth application en sélectionnant Créer une nouvelle connexion par jeton d'application Oauth dans la description.

Pour consulter vos OAuth applications autorisées, accédez à [Applications](#) activées GitHub et vérifiez qu'une application nommée AWS CodeBuild (*region*) appartenant à [aws-codesuite](#) est répertoriée.

## Accès à Bitbucket dans CodeBuild

Pour Bitbucket, vous utilisez un jeton d'accès, un mot de passe d'application, une OAuth application ou une connexion Bitbucket pour accéder au fournisseur source.

### Rubriques

- [Connexions à l'application Bitbucket](#)
- [Mot de passe ou jeton d'accès de l'application Bitbucket](#)
- [Application Bitbucket OAuth](#)

## Connexions à l'application Bitbucket

Vous pouvez utiliser Bitbucket pour vous connecter à CodeBuild. Les connexions à l'application Bitbucket sont prises en charge via [AWS CodeConnections](#).

### Note

CodeConnections est disponible dans moins de régions que CodeBuild. Vous pouvez utiliser des connexions entre régions dans CodeBuild. Les connexions créées dans les régions optionnelles ne peuvent pas être utilisées dans d'autres régions. Pour plus d'informations, consultez [Points de terminaison et quotas AWS CodeConnections](#).

### Rubriques

- [Étape 1 : créer une connexion à Bitbucket \(console\)](#)
- [Étape 2 : Accorder l'accès au rôle IAM du CodeBuild projet pour utiliser la connexion](#)
- [Étape 3 : Configuration CodeBuild pour utiliser la nouvelle connexion](#)

## Étape 1 : créer une connexion à Bitbucket (console)

Suivez ces étapes pour utiliser la CodeBuild console afin d'ajouter une connexion pour votre projet dans Bitbucket.

Pour créer une connexion à Bitbucket

- Suivez les instructions du guide de l'utilisateur des outils de développement pour [créer une connexion à Bitbucket](#).

### Note

Au lieu de créer ou d'utiliser une connexion existante dans votre compte, vous pouvez utiliser une connexion partagée depuis un autre AWS compte. Pour plus d'informations, voir [Partager des connexions avec AWS des comptes](#).

## Étape 2 : Accorder l'accès au rôle IAM du CodeBuild projet pour utiliser la connexion

Vous pouvez accorder l'accès au rôle IAM du CodeBuild projet pour utiliser les jetons Bitbucket vendus par votre connexion.

Pour accorder l'accès CodeBuild au rôle IAM du projet

1. Créez un rôle IAM pour votre CodeBuild projet en suivant les instructions correspondant [CodeBuild Autoriser l'interaction avec d'autres AWS services](#) à votre CodeBuild projet.
2. Tout en suivant les instructions, ajoutez la politique IAM suivante à votre rôle de CodeBuild projet pour autoriser l'accès à la connexion.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:StartBuild",
 "Resource": "arn:aws:codebuild:us-east-1:123456789012:project/MyProject:build/*"
```

```
{
 "Effect": "Allow",
 "Action": [
 "codeconnections:GetConnectionToken",
 "codeconnections:GetConnection"
],
 "Resource": [
 "arn:aws:iam::*:role/Service*"
]
}
```

### Étape 3 : Configuration CodeBuild pour utiliser la nouvelle connexion

Vous pouvez configurer une connexion en tant qu'identifiant au niveau du compte et l'utiliser dans un projet.

#### AWS Management Console

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS Management Console

1. Pour Source provider (Fournisseur de source), choisissez Bitbucket.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez Informations d'identification source par défaut pour utiliser les informations d'identification source par défaut de votre compte pour les appliquer à tous les projets.
    - a. Si vous n'êtes pas connecté à Bitbucket, choisissez Gérer les informations d'identification de source par défaut.
    - b. Pour le type d'identifiant, choisissez CodeConnections.
    - c. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.
  - Choisissez Identifiant source personnalisé pour utiliser un identifiant source personnalisé afin de remplacer les paramètres par défaut de votre compte.
    - a. Pour le type d'identifiant, choisissez CodeConnections.
    - b. Dans Connexion, choisissez d'utiliser une connexion existante ou d'en créer une nouvelle.

## AWS CLI

Pour configurer une connexion en tant qu'identifiant au niveau du compte dans AWS CLI

- Ouvrez une invite de terminal (Linux, macOS ou Unix) ou de commande (Windows). Utilisez le AWS CLI pour exécuter la `import-source-credentials` commande, en spécifiant le `--auth-type` `--server-type`, et `--token` pour votre connexion.

Utilisez la commande suivante :

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type BITBUCKET --token <connection-arn>
```

Pour plus d'informations sur la configuration de plusieurs jetons dans votre CodeBuild projet, consultez [Configurer plusieurs jetons comme informations d'identification au niveau de la source](#).

## Mot de passe ou jeton d'accès de l'application Bitbucket

### Conditions préalables

Avant de commencer, vous devez ajouter les étendues d'autorisation appropriées au mot de passe ou au jeton d'accès de votre application Bitbucket.

Pour Bitbucket, le mot de passe ou le jeton d'accès de votre application doit avoir les champs d'application suivants.

- `repository:read` : accorde l'accès en lecture à tous les référentiels auxquels l'utilisateur d'autorisation a accès.
- `pullrequest:read` : accorde l'accès en lecture aux demandes d'extraction. Si votre projet possède un webhook Bitbucket, le mot de passe ou le jeton d'accès de votre application doit avoir cette portée.
- `webhook` : accorde l'accès aux webhooks. Si votre projet comporte une opération webhook, le mot de passe ou le jeton d'accès de votre application doit avoir cette portée.
- `compte` : accorde un accès en lecture aux informations du compte de l'utilisateur.

Pour plus d'informations, consultez [Scopes for Bitbucket Cloud REST API](#) et Bitbucket [Cloud OAuth sur le site Web de Bitbucket](#).

## Connect Bitbucket avec un mot de passe d'application (console)

Pour utiliser la console afin de connecter votre projet à Bitbucket à l'aide d'un mot de passe d'application, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Pour Source provider (Fournisseur de source), choisissez Bitbucket.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez d'utiliser les informations d'identification du compte pour appliquer les informations d'identification source par défaut de votre compte à tous les projets.
    - a. Si vous n'êtes pas connecté à Bitbucket, choisissez Gérer les informations d'identification du compte.
    - b. Pour le type d'identifiant, choisissez le mot de passe de l'application.
  - Si vous avez choisi d'utiliser les informations d'identification au niveau du compte pour le service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :
    - a. Si vous choisissez d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou d'en créer un nouveau, puis choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
    - b. Si vous choisissez de l'utiliser CodeBuild, entrez votre nom d'utilisateur Bitbucket et le mot de passe de l'application, puis choisissez Enregistrer.
  - Sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement pour utiliser des informations d'identification source personnalisées afin de remplacer les paramètres d'identification de votre compte.
    - a. Dans la liste des informations d'identification renseignée, choisissez l'une des options sous Mot de passe de l'application.
    - b. Vous pouvez également créer un nouveau jeton de mot de passe d'application en sélectionnant créer une nouvelle connexion par mot de passe d'application dans la description.



## Connect Bitbucket avec un jeton d'accès (console)

Pour utiliser la console afin de connecter votre projet à Bitbucket à l'aide d'un jeton d'accès, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Pour Source provider (Fournisseur de source), choisissez Bitbucket.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez d'utiliser les informations d'identification du compte pour appliquer les informations d'identification source par défaut de votre compte à tous les projets.
    - a. Si vous n'êtes pas connecté à Bitbucket, choisissez Gérer les informations d'identification du compte.
    - b. Pour le type d'identifiant, choisissez jeton d'accès personnel.
  - Si vous avez choisi d'utiliser les informations d'identification au niveau du compte pour le service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :
    - a. Si vous choisissez d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou d'en créer un nouveau, puis choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
    - b. Si vous choisissez de l'utiliser CodeBuild, entrez votre jeton d'accès personnel Bitbucket, puis choisissez Enregistrer.
  - Sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement pour utiliser des informations d'identification source personnalisées afin de remplacer les paramètres d'identification de votre compte.
    - a. Dans la liste des informations d'identification renseignée, choisissez l'une des options sous Jeton d'accès personnel.
    - b. Vous pouvez également créer un nouveau jeton d'accès personnel en sélectionnant créer une nouvelle connexion par jeton d'accès personnel dans la description.

Connectez Bitbucket à l'aide d'un mot de passe d'application ou d'un jeton d'accès (CLI)

Suivez ces étapes pour connecter votre projet à Bitbucket AWS CLI à l'aide d'un mot de passe d'application ou d'un jeton d'accès. Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

1. Exécutez la commande `import-source-credentials` :

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, *import-source-credentials.json*) situé sur l'ordinateur local ou sur l'instance où le AWS CLI est installé. Modifiez les données copiées comme suit, puis enregistrez vos résultats.

```
{
 "serverType": "BITBUCKET",
 "authType": "auth-type",
 "shouldOverwrite": "should-overwrite",
 "token": "token",
 "username": "username"
}
```

Remplacez les éléments suivants :

- *server-type*: valeur requise. Le fournisseur de source utilisé pour cette information d'identification. Les valeurs valides sont GITHUB, BITBUCKET, GITHUB\_ENTERPRISE, GITLAB et GITLAB\_SELF\_MANAGED.
- *auth-type*: valeur requise. Type d'authentification utilisé pour se connecter à un référentiel. Les valeurs valides sont OAUTH, BASIC\_AUTH, PERSONAL\_ACCESS\_TOKEN, CODECONNECTIONS et SECRETS\_MANAGER. Car GitHub seul PERSONAL\_ACCESS\_TOKEN est autorisé. BASIC\_AUTH n'est autorisé qu'avec le mot de passe de l'application Bitbucket.
- *should-overwrite*: valeur facultative. Définissez cette valeur sur `false` pour empêcher le remplacement des informations d'identification source du référentiel. Définissez cette valeur sur `true` pour remplacer les informations d'identification source du référentiel. La valeur par défaut est `true`.

- **token**: valeur requise. Pour GitHub notre serveur d' GitHub entreprise, il s'agit du jeton d'accès personnel. Pour Bitbucket, il s'agit du jeton d'accès personnel ou du mot de passe de l'application. Pour le type d'authentification CODECONNECTIONS, il s'agit de l'ARN de connexion. Pour le type d'authentification SECRETS\_MANAGER, il s'agit de l'ARN secret.
  - **username**: valeur facultative. Ce paramètre est ignoré pour GitHub les fournisseurs de sources GitHub Enterprise Server.
2. Pour connecter votre compte à l'aide d'un mot de passe d'application ou d'un jeton d'accès, accédez au répertoire qui contient le `import-source-credentials.json` fichier que vous avez enregistré à l'étape 1 et réexécutez la `import-source-credentials` commande.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Les données au format JSON apparaissent dans la sortie avec un Amazon Resource Name (ARN).

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

#### Note

Si vous exécutez la commande `import-source-credentials` avec le même type de serveur et de auth une seconde fois, le jeton d'accès stocké est mis à jour.

Une fois que votre compte est connecté à l'aide d'un mot de passe d'application, vous pouvez `create-project` l'utiliser pour créer votre CodeBuild projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

3. Pour afficher les mots de passe ou les jetons d'accès des applications connectées, exécutez la `list-source-credentials` commande.

```
aws codebuild list-source-credentials
```

Un objet `sourceCredentialsInfos` au format JSON apparaît dans la sortie :

```
{
 "sourceCredentialsInfos": [
 {
 "authType": "auth-type",
 "serverType": "BITBUCKET",
 "arn": "arn"
 }
]
}
```

Le `sourceCredentialsObject` contient la liste des informations d'identification de la source connectée :

- L'élément `authType` est le type d'authentification utilisé par les informations d'identification. Cela peut être `OAUTHBASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS`, ou `SECRETS_MANAGER`.
  - Le `serverType` est le type de fournisseur de source. Cela peut être `GITHUBGITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB`, ou `GITLAB_SELF_MANAGED`.
  - Le `arn` est l'ARN du jeton.
4. Pour vous déconnecter d'un fournisseur source et supprimer son mot de passe d'application ou ses jetons d'accès, exécutez la `delete-source-credentials` commande avec son ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Des données au format JSON sont renvoyées avec un ARN des informations d'identification supprimées.

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

## Application Bitbucket OAuth

### Connect Bitbucket en utilisant OAuth (console)

Pour utiliser la console afin de connecter votre projet à Bitbucket à l'aide d'une OAuth application, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Pour Source provider (Fournisseur de source), choisissez Bitbucket.
2. Pour Credential, effectuez l'une des opérations suivantes :
  - Choisissez d'utiliser les informations d'identification du compte pour appliquer les informations d'identification source par défaut de votre compte à tous les projets.
    - a. Si vous n'êtes pas connecté à Bitbucket, choisissez Gérer les informations d'identification du compte.
    - b. Pour le type d'identifiant, choisissez OAuth app.
  - Si vous avez choisi d'utiliser les informations d'identification au niveau du compte pour le service, choisissez le service que vous souhaitez utiliser pour stocker votre jeton et procédez comme suit :
    - a. Si vous choisissez d'utiliser Secrets Manager, vous pouvez choisir d'utiliser une connexion secrète existante ou d'en créer un nouveau, puis choisir Enregistrer. Pour plus d'informations sur la création d'un nouveau secret, consultez [Créez et stockez un jeton dans un secret du Gestionnaire de Secrets](#).
    - b. Si vous choisissez d'utiliser, CodeBuild puis choisissez Enregistrer.
  - Sélectionnez Utiliser les informations d'identification de remplacement pour ce projet uniquement pour utiliser des informations d'identification source personnalisées afin de remplacer les paramètres d'identification de votre compte.
    - a. Dans la liste des informations d'identification renseignée, choisissez l'une des options sous OAuth application.
    - b. Vous pouvez également créer un nouveau jeton d' OAuth application en sélectionnant Créer une nouvelle connexion par jeton d'application Oauth dans la description.

Pour consulter vos OAuth applications autorisées, accédez à [Autorisations d'applications](#) sur Bitbucket et vérifiez qu'une application nommée AWS CodeBuild (*region*) est répertoriée.

# GitLab accéder à CodeBuild

En GitLab effet, vous utilisez une GitLab connexion pour accéder au fournisseur source.

## Rubriques

- [Connect CodeBuild à GitLab](#)

## Connect CodeBuild à GitLab

Les connexions vous permettent d'autoriser et d'établir des configurations qui associent votre fournisseur tiers aux AWS ressources que vous utilisez AWS CodeConnections. Pour associer votre référentiel tiers en tant que source pour votre projet de construction, vous utilisez une connexion.

Pour ajouter un fournisseur de source GitLab ou un fournisseur de source GitLab autogéré CodeBuild, vous pouvez choisir de :

- Utilisez l'assistant de création de projet ou la page Modifier la source de la CodeBuild console pour choisir l'option GitLab ou le fournisseur GitLab autogéré. Consultez [Création d'une connexion à GitLab \(console\)](#) la section pour ajouter le fournisseur source. La console vous permet de créer une ressource de connexions.
- Utilisez la CLI pour créer vos ressources de connexions, voir [Création d'une connexion à GitLab \(CLI\)](#) pour créer une ressource de connexions avec la CLI.

### Note

Vous pouvez également créer une connexion à l'aide de la console Developer Tools sous Paramètres. Voir [Créer une connexion](#).

### Note

En autorisant l'installation de cette connexion GitLab, vous autorisez notre service à traiter vos données en accédant à votre compte, et vous pouvez révoquer ces autorisations à tout moment en désinstallant l'application.

## Créez une connexion avec GitLab

Cette section décrit comment se connecter GitLab à CodeBuild. Pour plus d'informations sur les connexions GitLab , consultez [Connect CodeBuild à GitLab](#).

Avant de commencer :

- Vous devez déjà avoir créé un compte auprès de GitLab.

### Note

Les connexions fournissent uniquement l'accès aux référentiels appartenant au compte qui a été utilisé pour créer et autoriser la connexion.

### Note

Vous pouvez créer des connexions à un référentiel dans lequel vous avez le rôle de propriétaire GitLab, puis la connexion peut être utilisée avec le référentiel avec des ressources telles que CodeBuild. Pour les référentiels dans des groupes, il n'est pas nécessaire d'être le propriétaire du groupe.

- Pour spécifier une source pour votre projet de build, vous devez déjà avoir créé un référentiel sur GitLab.

## Rubriques

- [Création d'une connexion à GitLab \(console\)](#)
- [Création d'une connexion à GitLab \(CLI\)](#)

## Création d'une connexion à GitLab (console)

Suivez ces étapes pour utiliser la CodeBuild console afin d'ajouter une connexion pour votre projet (référentiel) dans GitLab.

 Note

Au lieu de créer ou d'utiliser une connexion existante dans votre compte, vous pouvez utiliser une connexion partagée depuis un autre AWS compte. Pour plus d'informations, voir [Partager des connexions avec AWS des comptes](#).

Pour créer ou modifier votre projet de construction

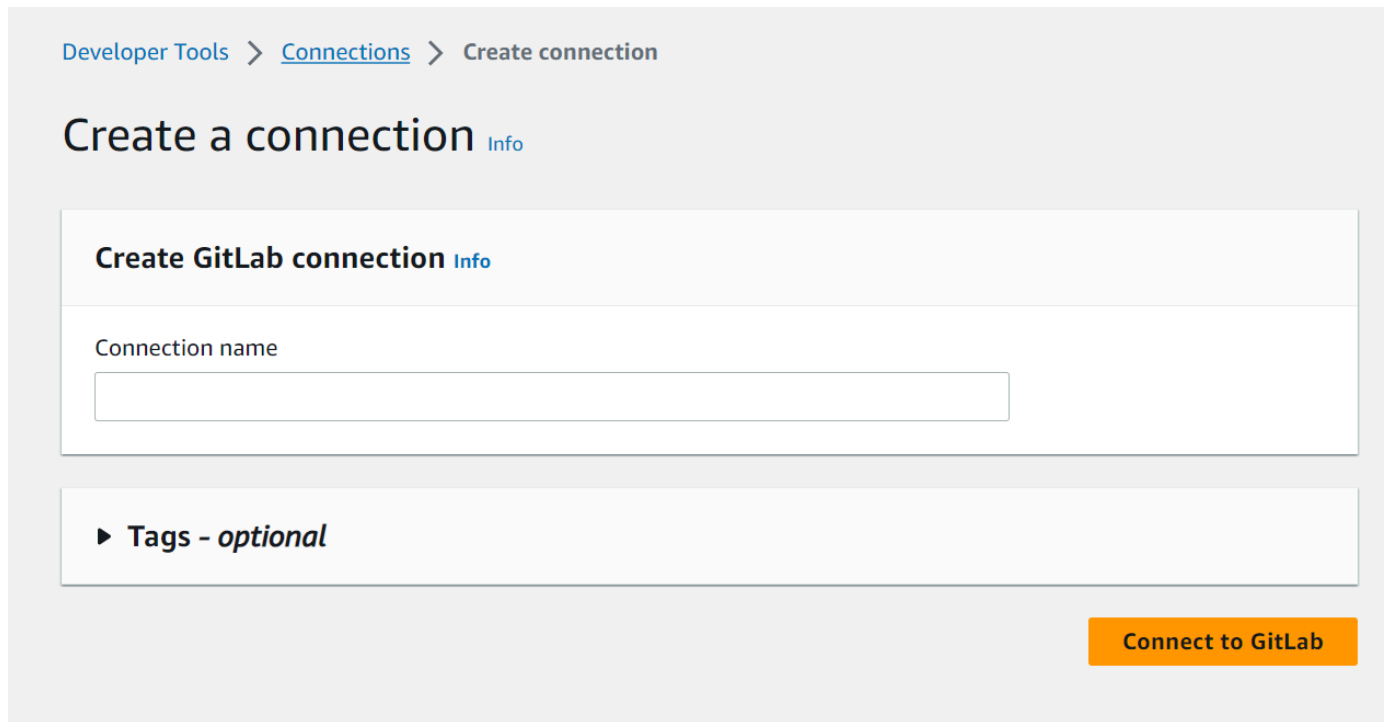
1. Connectez-vous à la CodeBuild console.
2. Choisissez l'une des options suivantes.
  - Choisissez de créer un projet de construction. Suivez les étapes décrites [Création d'un projet de génération \(console\)](#) pour terminer le premier écran et dans la section Source, sous Source Provider, sélectionnez GitLab.
  - Choisissez de modifier un projet de construction existant. Choisissez Modifier, puis Source. Sur la page Modifier la source, sous Fournisseur de source, sélectionnez GitLab.
3. Sélectionnez l'une des méthodes suivantes :
  - Sous Connexion, choisissez Connexion par défaut. La connexion par défaut applique une GitLab connexion par défaut à tous les projets.
  - Sous Connexion, choisissez Connexion personnalisée. Connexion personnalisée applique une GitLab connexion personnalisée qui remplace les paramètres par défaut de votre compte.
4. Effectuez l'une des actions suivantes :
  - Sous Connexion par défaut ou Connexion personnalisée, si vous n'avez pas encore créé de connexion avec votre fournisseur, choisissez Créer une nouvelle GitLab connexion. Passez à l'étape 5 pour créer la connexion.
  - Sous Connexion, si vous avez déjà créé une connexion avec votre fournisseur, choisissez-la. Passez à l'étape 10.

 Note

Si vous fermez la fenêtre contextuelle avant de créer une GitLab connexion, vous devez actualiser la page.



5. Pour créer une connexion à un GitLab référentiel, sous Sélectionnez un fournisseur, sélectionnez GitLab. Dans Connection name (Nom de la connexion), saisissez le nom de la connexion que vous souhaitez créer. Choisissez Connect to (Se connecter à) GitLab.



The screenshot shows the 'Create a connection' page in the AWS CodeBuild console. At the top, there is a breadcrumb trail: 'Developer Tools > Connections > Create connection'. Below this, the main heading is 'Create a connection' with an 'Info' link. The page contains a form with a section titled 'Create GitLab connection' with an 'Info' link. Inside this section, there is a label 'Connection name' above a text input field. Below the input field, there is a section titled 'Tags - optional' with a right-pointing triangle icon. At the bottom right of the form, there is an orange button labeled 'Connect to GitLab'.

6. Lorsque la page de connexion GitLab s'affiche, connectez-vous avec vos informations d'identification, puis choisissez Se connecter.
7. Si c'est la première fois que vous autorisez la connexion, une page d'autorisation s'affiche avec un message demandant l'autorisation de connexion pour accéder à votre GitLab compte.

Choisissez Authorize (Autoriser).

## Authorize **AWS Connector for GitLab** to use your account?

An application called **AWS Connector for GitLab** is requesting access to your GitLab account. This application was created by **Amazon AWS**. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**

Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.

- **Read the authenticated user's personal information**

Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

- **Read Api**

Grants read access to the API, including all groups and projects, the container registry, and the package registry.

- **Allows read-only access to the repository**

Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.

- **Allows read-write access to the repository**

Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

8. Le navigateur revient à la page de la console des connexions. Dans les paramètres de GitLab connexion, la nouvelle connexion est affichée dans Nom de la connexion.
9. Choisissez Se connecter.

Une fois la GitLab connexion créée avec succès, une bannière de réussite s'affiche en haut de la page.

10. Sur la page Créer un projet de construction, dans la liste déroulante Connexion par défaut ou Connexion personnalisée, assurez-vous que l'ARN de votre connexion est répertorié. Sinon, cliquez sur le bouton d'actualisation pour le faire apparaître.
11. Dans Repository, choisissez le nom de votre projet en GitLab spécifiant le chemin du projet avec l'espace de noms. Par exemple, pour un référentiel au niveau du groupe, entrez le nom du référentiel au format suivant : `group-name/repository-name` Pour plus d'informations sur le chemin et l'espace de noms, consultez le `path_with_namespace` champ dans <https://docs.gitlab.com/ee/get-single-projectapi/projects.html#>. Pour plus d'informations sur l'espace de noms dans GitLab, consultez <https://docs.gitlab.com/ee/user/namespace/>.

 Note

Pour les groupes dans GitLab, vous devez spécifier manuellement le chemin du projet avec l'espace de noms. Par exemple, pour un référentiel nommé `myrepo` dans un groupe `mygroup`, entrez ce qui suit : `mygroup/myrepo`. Vous pouvez trouver le chemin du projet avec l'espace de noms dans l'URL dans GitLab.

12. Dans Version source (facultatif), entrez un identifiant de pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour de plus amples informations, veuillez consulter [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des validations IDs, tels que `811dd1ba1aba14473856cee38308caed7190c0d` ou `5392f7`. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

13. Dans Git clone depth (facultatif), vous pouvez créer un clone superficiel dont l'historique est tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.
14. Dans État de la construction - facultatif, sélectionnez Signaler les statuts de construction au fournisseur source lorsque vos builds commencent et se terminent si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

### Création d'une connexion à GitLab (CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer une connexion.

Pour ce faire, utilisez la commande create-connection.

#### Important

Une connexion créée via le AWS CLI ou AWS CloudFormation est en PENDING état par défaut. Après avoir créé une connexion avec la CLI CloudFormation, utilisez la console pour modifier la connexion afin de définir son étatAVAILABLE.

Pour créer une connexion

- Suivez les instructions du guide de l'utilisateur de la console Developer Tools pour [créer une connexion à GitLab \(CLI\)](#).

## Prévention du cas de figure de l'adjoind désorienté entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé à accéder.

autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS CodeBuild accordent un autre service à la ressource. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (\*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:codebuild:*:123456789012:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations.

La valeur de `aws:SourceArn` doit être l'ARN CodeBuild du projet.

L'exemple suivant montre comment utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale CodeBuild pour éviter le problème de confusion des adjoints.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": "arn:aws:codebuild:us-east-1:111122223333:project/MyProject"
 }
 }
 }
]
}
```

```
}
]
 }
 }
 }
```

# Rubriques avancées

Cette section inclut plusieurs rubriques avancées qui s'avèrent utiles pour des utilisateurs AWS CodeBuild plus expérimentés.

## Rubriques

- [Permettre aux utilisateurs d'interagir avec CodeBuild](#)
- [CodeBuild Autoriser l'interaction avec d'autres AWS services](#)
- [Chiffrez les résultats de construction à l'aide d'une clé gérée par le client](#)
- [Interagissez à CodeBuild l'aide du AWS CLI](#)
- [Référence de ligne de commande pour AWS CodeBuild](#)
- [AWS SDKs et référence d'outils pour AWS CodeBuild](#)
- [Utilisation de ce service avec un AWS SDK](#)
- [Spécifiez le AWS CodeBuild point de terminaison](#)
- [Utilisez AWS CodeBuild with AWS CodePipeline pour tester le code et exécuter des builds](#)
- [Utilisation AWS CodeBuild avec Codecov](#)
- [À utiliser AWS CodeBuild avec Jenkins](#)
- [Utilisation AWS CodeBuild avec des applications sans serveur](#)
- [Notices émanant de tiers AWS CodeBuild pour Windows](#)
- [Utiliser des clés de CodeBuild condition comme variables de rôle de service IAM pour contrôler l'accès au build](#)
- [AWS CodeBuild clés de condition](#)

## Permettre aux utilisateurs d'interagir avec CodeBuild

Si vous suivez les étapes décrites [Mise en route à l'aide de la console](#) pour y accéder AWS CodeBuild pour la première fois, vous n'avez probablement pas besoin des informations contenues dans cette rubrique. Toutefois, au fur et à mesure de votre utilisation CodeBuild, vous souhaitez peut-être, par exemple, donner aux autres utilisateurs et groupes de votre organisation la possibilité d'interagir avec eux CodeBuild.

Pour permettre à un utilisateur ou à un groupe IAM d'interagir avec AWS CodeBuild, vous devez lui accorder des autorisations d'accès à CodeBuild. Cette section décrit comment procéder avec la console IAM ou le AWS CLI.

Si vous souhaitez accéder CodeBuild à votre compte AWS root (ce n'est pas recommandé) ou à l'aide d'un utilisateur administrateur de votre AWS compte, vous n'avez pas besoin de suivre ces instructions.

Pour plus d'informations sur les comptes AWS root et les utilisateurs administrateurs, voir [L'utilisateur Compte AWS root](#) et [Création de votre premier utilisateur et groupe Compte AWS root](#) dans le guide de l'utilisateur.

Pour ajouter des autorisations CodeBuild d'accès à un groupe ou à un utilisateur IAM (console)

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Vous devez déjà vous être connecté au en AWS Management Console utilisant l'une des méthodes suivantes :

- Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, voir [L'utilisateur Compte AWS root](#) dans le guide de l'utilisateur.
- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam>ListAttachedGroupPolicies
iam>ListAttachedUserPolicies
iam>ListGroups
iam>ListPolicies
iam>ListUsers
```

Pour plus d'informations, consultez la section [Présentation des politiques IAM](#) dans le guide de l'utilisateur.

2. Dans le panneau de navigation, choisissez Politiques.
3. Pour ajouter un ensemble personnalisé d'autorisations d' AWS CodeBuild accès à un groupe IAM ou à un utilisateur IAM, passez à l'étape 4 de cette procédure.



Pour ajouter un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe IAM ou à un utilisateur IAM, choisissez Type de stratégie, AWS Géré, puis procédez comme suit :

- Pour ajouter des autorisations d'accès complètes à CodeBuild, cochez la case nommée **AWSCodeBuildAdminAccess**, choisissez Actions de politique, puis choisissez Joindre. Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette opération pour les politiques nommées AmazonS3 ReadOnlyAccess et IAMFull Access.
- Pour ajouter des autorisations d'accès CodeBuild pour tout sauf pour l'administration du projet de création, cochez la case nommée **AWSCodeBuildDeveloperAccess**, choisissez Actions politiques, puis choisissez Joindre. Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette opération pour la politique nommée AmazonS3 ReadOnlyAccess.
- Pour ajouter des autorisations d'accès en lecture seule à CodeBuild, cochez les cases nommées. **AWSCodeBuildReadOnlyAccess** Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette opération pour la politique nommée AmazonS3 ReadOnlyAccess.

Vous avez maintenant ajouté un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe ou à un utilisateur IAM. Ignorez les autres étapes de cette procédure.

4. Choisissez Create Policy (Créer une politique).
5. Sur la page Créer une stratégie, choisissez Sélectionner en regard de Créer votre propre stratégie.
6. Sur la page Review Policy (Examiner la stratégie), pour Policy Name (Nom de la stratégie), saisissez un nouveau nom de stratégie (par exemple, **CodeBuildAccessPolicy**). Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.
7. Dans Policy Document (Document de stratégie), saisissez ce qui suit, puis sélectionnez Create Policy (Créer une stratégie).

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildAccessPolicy",
```

```

 "Effect": "Allow",
 "Action": [
 "codebuild:*"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeBuildRolePolicy",
 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::111122223333:role/role-name"
 },
 {
 "Sid": "CloudWatchLogsAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:FilterLogEvents",
 "logs:GetLogEvents"
],
 "Resource": "*"
 },
 {
 "Sid": "S3AccessPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:CreateBucket",
 "s3:GetObject",
 "s3:List*",
 "s3:PutObject"
],
 "Resource": "*"
 },
 {
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
 }
]

```

```
}
```

**Note**

Cette politique permet d'accéder à toutes les CodeBuild actions et à un nombre potentiellement important de AWS ressources. Pour restreindre les autorisations à des CodeBuild actions spécifiques, modifiez la valeur de `codebuild:*` dans la déclaration CodeBuild de politique. Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès](#). Pour restreindre l'accès à des AWS ressources spécifiques, modifiez la valeur de `Resource`. Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès](#).

8. Dans le volet de navigation, choisissez Groupes ou Utilisateurs.
9. Dans la liste des groupes ou des utilisateurs, choisissez le nom du groupe IAM ou de l'utilisateur IAM auquel vous souhaitez ajouter des autorisations d' CodeBuild accès.
10. Pour un groupe, sur la page des paramètres du groupe, dans l'onglet Permissions (Autorisations), développez Managed Policies (Stratégies gérées), puis choisissez Attach Policy (Attacher une stratégie).

Pour un utilisateur, sur la page des paramètres de l'utilisateur, dans l'onglet Autorisations, choisissez Ajouter des autorisations.

11. Pour un groupe, sur la page Joindre une politique, sélectionnez CodeBuildAccessPolicy, puis choisissez Attacher une politique.

Pour un utilisateur, sur la page Ajouter des autorisations, choisissez Joindre directement les politiques existantes. Sélectionnez CodeBuildAccessPolicy, choisissez Suivant : Révision, puis choisissez Ajouter des autorisations.

Pour ajouter des autorisations CodeBuild d'accès à un groupe ou à un utilisateur IAM (AWS CLI)

1. Assurez-vous d'avoir configuré la clé AWS CLI d' AWS accès et la clé d'accès AWS secrète correspondant à l'une des entités IAM, comme décrit dans la procédure précédente. Pour plus d'informations, consultez [Préparation de l'installation de l' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface .
2. Pour ajouter un ensemble personnalisé d'autorisations d' AWS CodeBuild accès à un groupe ou à un utilisateur IAM, passez à l'étape 3 de cette procédure.

Pour ajouter un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe ou à un utilisateur IAM, procédez comme suit :

Exécutez l'une des commandes suivantes, selon que vous souhaitez ajouter des autorisations à un groupe ou à un utilisateur IAM :

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn

aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Vous devez exécuter la commande trois fois, en *user-name* remplaçant *group-name* ou par le nom du groupe IAM ou le nom d'utilisateur, et en remplaçant *policy-arn* une fois pour chacune des politiques Amazon Resource Names (ARNs) suivantes :

- Pour ajouter des autorisations d'accès complètes à CodeBuild, appliquez la politique suivante ARNs :
  - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
  - `arn:aws:iam::aws:policy/IAMFullAccess`
- Pour ajouter des autorisations d'accès CodeBuild pour tout sauf pour l'administration du projet de build, appliquez la politique suivante ARNs :
  - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Pour ajouter des autorisations d'accès en lecture seule à CodeBuild, appliquez la politique suivante : ARNs
  - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Vous avez maintenant ajouté un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe ou à un utilisateur IAM. Ignorez les autres étapes de cette procédure.

3. Dans un répertoire vide du poste de travail local ou de l'instance où le AWS CLI est installé, créez un fichier nommé `put-group-policy.json` ou `put-user-policy.json`. Si vous choisissez un autre nom de fichier, veillez à l'utiliser tout au long de cette procédure.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "codebuild:*"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeBuildRolePolicy",
 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::111122223333:role/role-name"
 },
 {
 "Sid": "CloudWatchLogsAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:FilterLogEvents",
 "logs:GetLogEvents"
],
 "Resource": "*"
 },
 {
 "Sid": "S3AccessPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:CreateBucket",
 "s3:GetObject",
 "s3:List*",
 "s3:PutObject"
],
 "Resource": "*"
 }
]
}
```

```
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
 }
}
```

#### Note

Cette politique permet d'accéder à toutes les CodeBuild actions et à un nombre potentiellement important de AWS ressources. Pour restreindre les autorisations à des CodeBuild actions spécifiques, modifiez la valeur de `codebuild:*` dans la déclaration CodeBuild de politique. Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès](#). Pour restreindre l'accès à des AWS ressources spécifiques, modifiez la valeur de l'objet `Resource` associé. Pour plus d'informations, consultez [Gestion des identités et des accès](#) ou la documentation de sécurité du service AWS spécifique.

4. Placez-vous dans le répertoire dans lequel vous avez enregistré le fichier, puis exécutez l'une des commandes suivantes. Vous pouvez utiliser des valeurs différentes pour `CodeBuildGroupAccessPolicy` et `CodeBuildUserAccessPolicy`. Si vous utilisez des valeurs différentes, substituez-les ici.

Pour un groupe IAM :

```
aws iam put-group-policy --group-name group-name --policy-name
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Pour un utilisateur :

```
aws iam put-user-policy --user-name user-name --policy-name
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

Dans les commandes précédentes, remplacez *group-name* ou *user-name* par le nom du groupe ou de l'utilisateur IAM cible.

## CodeBuild Autoriser l'interaction avec d'autres AWS services

Si vous suivez les étapes décrites [Mise en route à l'aide de la console](#) pour y accéder AWS CodeBuild pour la première fois, vous n'avez probablement pas besoin des informations contenues dans cette rubrique. Cependant, au fur et à mesure que vous continuez à utiliser CodeBuild, vous souhaitez peut-être faire des choses telles que CodeBuild autoriser l'interaction avec d'autres AWS services.

CodeBuild Pour pouvoir interagir avec les AWS services dépendants en votre nom, vous avez besoin d'un rôle AWS CodeBuild de service. Vous pouvez créer un rôle CodeBuild de service à l'aide des AWS CodePipeline consoles CodeBuild or. Pour plus d'informations, consultez :

- [Création d'un projet de génération \(console\)](#)
- [Créez un pipeline qui utilise CodeBuild \(CodePipeline console\)](#)
- [Ajouter une CodeBuild action de génération à un pipeline \(CodePipeline console\)](#)
- [Modification des paramètres d'un projet de génération \(console\)](#)

Si vous ne prévoyez pas d'utiliser ces consoles, cette section explique comment créer un rôle de CodeBuild service avec la console IAM ou le AWS CLI.

### Important

CodeBuild utilise le rôle de service pour toutes les opérations effectuées en votre nom. Si le rôle comprend des autorisations que l'utilisateur ne devrait pas avoir, vous avez peut-être remonté accidentellement ses autorisations. Vérifiez que le rôle accorde le [privilège le plus faible](#).

Le rôle de service décrit sur cette page contient une stratégie qui accorde les autorisations minimales requises pour utiliser CodeBuild. Il se peut que vous deviez ajouter des autorisations supplémentaires, en fonction de votre cas d'utilisation.

## Pour créer un rôle CodeBuild de service (console)

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Vous devez vous être déjà connecté à la console à l'aide de l'une des actions suivantes :

- Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, voir [L'utilisateur Compte AWS root](#) dans le guide de l'utilisateur.
- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam:ListAttachedRolePolicies
iam:ListPolicies
iam:ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Pour plus d'informations, consultez la section [Présentation des politiques IAM](#) dans le guide de l'utilisateur.

2. Dans le panneau de navigation, choisissez Politiques.
3. Choisissez Create Policy (Créer une politique).
4. Sur la page Créer une stratégie, choisissez JSON.
5. Pour la stratégie JSON, saisissez ce qui suit, puis choisissez Review Policy (Examiner une stratégie) :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
```



```
{
 "Sid": "CloudWatchLogsPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
 "Resource": "*"
},
{
 "Sid": "CodeCommitPolicy",
 "Effect": "Allow",
 "Action": [
 "codecommit:GitPull"
],
 "Resource": "*"
},
{
 "Sid": "S3GetObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*"
},
{
 "Sid": "S3PutObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
],
 "Resource": "*"
},
{
 "Sid": "ECRPullPolicy",
 "Effect": "Allow",
 "Action": [
 "ecr:BatchCheckLayerAvailability",
 "ecr:GetDownloadUrlForLayer",
 "ecr:BatchGetImage"
],
 "Resource": "*"
}
```

```
 },
 {
 "Sid": "ECRAuthPolicy",
 "Effect": "Allow",
 "Action": [
 "ecr:GetAuthorizationToken"
],
 "Resource": "*"
 },
 {
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
 }
]
}
```

#### Note

Cette politique contient des déclarations qui autorisent l'accès à un nombre potentiellement important de AWS ressources. Pour restreindre AWS CodeBuild l'accès à des AWS ressources spécifiques, modifiez la valeur du Resource tableau. Pour plus d'informations, consultez la documentation de sécurité du AWS service.

6. Sur la page Review Policy (Examiner une stratégie), pour Policy Name (Nom de stratégie), saisissez un nom pour la stratégie (par exemple, **CodeBuildServiceRolePolicy**), puis choisissez Create policy (Créer une stratégie).

#### Note

Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.

7. Dans le panneau de navigation, choisissez Rôles.
8. Sélectionnez Create role (Créer un rôle).

9. Sur la page Créer un rôle, le AWS service étant déjà sélectionné, choisissez CodeBuild, puis choisissez Next:Permissions.
10. Sur la page Joindre des politiques d'autorisation CodeBuildServiceRolePolicy, sélectionnez, puis choisissez Suivant : Révision.
11. Sur la page Create role and review (Créer un rôle et vérifier), pour Role name (Nom du rôle), saisissez le nom du rôle (par exemple, **CodeBuildServiceRole**), puis choisissez Create role (Créer un rôle).

Pour créer un rôle CodeBuild de service (AWS CLI)

1. Assurez-vous d'avoir configuré la clé AWS CLI d' AWS accès et la clé d'accès AWS secrète correspondant à l'une des entités IAM, comme décrit dans la procédure précédente. Pour plus d'informations, consultez [Préparation de l' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface .
2. Dans un répertoire vide du poste de travail local ou de l'instance où le AWS CLI est installé, créez deux fichiers nommés `create-role.json` et `put-role-policy.json`. Si vous choisissez des noms de fichier différents, substituez-les tout au long de cette procédure.

`create-role.json`:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

 Note

Nous vous recommandons d'utiliser les clés de condition `aws:SourceAccount` et `aws:SourceArn` pour vous protéger contre [le problème du député confus](#). Par exemple, vous pouvez modifier la politique de confiance précédente avec les blocs de conditions suivants. `aws:SourceAccount` Il s'agit du propriétaire du CodeBuild projet et de `aws:SourceArn` l'ARN CodeBuild du projet.

Si vous souhaitez limiter votre rôle de service à un AWS compte, cela `create-role.json` pourrait ressembler à ceci :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": [
 "111122223333"
]
 }
 }
 }
]
}
```

Si vous souhaitez limiter votre rôle de service à un CodeBuild projet spécifique, cela `create-role.json` pourrait ressembler à ceci :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": "arn:aws:codebuild:us-east-1:111122223333:project/MyProject"
 }
 }
 }
]
}
```

 Note

Si vous ne connaissez pas ou n'avez pas encore choisi le nom de votre CodeBuild projet et que vous souhaitez une restriction de politique de confiance sur un modèle d'ARN particulier, vous pouvez remplacer cette partie de l'ARN par un caractère générique (\*). Après avoir créé votre projet, vous pouvez mettre à jour la politique de confiance.

put-role-policy.json:

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CloudWatchLogsPolicy",
```

```
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
 "Resource": "*"
},
{
 "Sid": "CodeCommitPolicy",
 "Effect": "Allow",
 "Action": [
 "codecommit:GitPull"
],
 "Resource": "*"
},
{
 "Sid": "S3GetObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*"
},
{
 "Sid": "S3PutObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
],
 "Resource": "*"
},
{
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
}
]
```

```
}
```

#### Note

Cette politique contient des déclarations qui autorisent l'accès à un nombre potentiellement important de AWS ressources. Pour restreindre AWS CodeBuild l'accès à des AWS ressources spécifiques, modifiez la valeur du Resource tableau. Pour plus d'informations, consultez la documentation de sécurité du AWS service.

3. Placez-vous dans le répertoire dans lequel vous avez enregistré les fichiers précédents, puis exécutez les deux commandes suivantes, une par une, dans cet ordre. Vous pouvez utiliser des valeurs différentes pour CodeBuildServiceRole et CodeBuildServiceRolePolicy, mais veillez à les substituer ici.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

## Chiffrez les résultats de construction à l'aide d'une clé gérée par le client

Si vous suivez les étapes décrites [Mise en route à l'aide de la console](#) pour y accéder AWS CodeBuild pour la première fois, vous n'avez probablement pas besoin des informations contenues dans cette rubrique. Cependant, au fur et à mesure que vous continuez à utiliser CodeBuild, vous souhaitez peut-être effectuer des opérations telles que chiffrer les artefacts de construction.

AWS CodeBuild Pour chiffrer ses artefacts de sortie de build, il doit avoir accès à une clé KMS. Par défaut, CodeBuild utilise le Clé gérée par AWS pour Amazon S3 dans votre AWS compte.

Si vous ne souhaitez pas utiliser le Clé gérée par AWS, vous devez créer et configurer vous-même une clé gérée par le client. Cette section décrit comment procéder avec la console IAM.

Pour plus d'informations sur les clés gérées par le client, consultez la section [AWS Key Management Service Concepts](#) et [création de clés](#) dans le guide du AWS KMS développeur.

Pour configurer une clé gérée par le client à utiliser par CodeBuild, suivez les instructions de la section « Comment modifier une politique clé » de la section [Modification d'une politique clé](#) du guide du AWS KMS développeur. Ajoutez ensuite les instructions suivantes (entre **### BEGIN ADDING STATEMENTS HERE ###** et **### END ADDING STATEMENTS HERE ###**) à la politique clé. Des ellipses (...) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter les instructions. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie de clé.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "kms:ViaService": "s3.us-east-1.amazonaws.com",
 "kms:CallerAccount": "111122223333"
 }
 }
 },
 {
 "Effect": "Allow",
 "Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey"
],
 "Resource": "*"
 }
]
}
```



```
}
```

- ***region-ID*** représente l'ID de la AWS région où se trouvent les compartiments Amazon S3 associés CodeBuild (par exemple, `us-east-1`).
- ***account-ID*** représente l'ID du AWS compte propriétaire de la clé gérée par le client.
- ***CodeBuild-service-role*** représente le nom du rôle de CodeBuild service que vous avez créé ou identifié précédemment dans cette rubrique.

#### Note

Pour créer ou configurer une clé gérée par le client via la console IAM, vous devez d'abord vous connecter au en AWS Management Console utilisant l'une des méthodes suivantes :

- Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, voir [The Account Root User](#) dans le guide de l'utilisateur.
- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à créer ou à modifier la clé gérée par le client. Pour plus d'informations, consultez la section [Autorisations requises pour utiliser la AWS KMS console](#) dans le guide du AWS KMS développeur.

## Interagissez à CodeBuild l'aide du AWS CLI

Si vous suivez les étapes décrites [Mise en route à l'aide de la console](#) pour y accéder AWS CodeBuild pour la première fois, vous n'avez probablement pas besoin des informations contenues dans cette rubrique. Toutefois, au fur et à mesure que vous continuez à utiliser CodeBuild, vous souhaitez peut-être faire des choses telles que permettre AWS CLI aux utilisateurs d'utiliser le pour interagir avec CodeBuild la CodeBuild console, la CodePipeline console ou le AWS SDKs.

Pour l'installer et le configurer AWS CLI, reportez-vous à la section [Getting Up with the AWS Command Line Interface](#) dans le guide de AWS Command Line Interface l'utilisateur.

Après avoir installé le AWS CLI, effectuez les tâches suivantes :

1. Exécutez la commande suivante pour vérifier si vous avez installé les AWS CLI supports CodeBuild :

```
aws codebuild list-builds
```

En cas de réussite, des informations similaires à ce qui suit s'affichent dans la sortie :

```
{
 "ids": []
}
```

Les crochets vides indiquent que vous n'avez pas encore exécuté de générations.

2. Si une erreur est générée, vous devez désinstaller votre version actuelle de l' AWS CLI , puis installer la dernière version. Pour plus d'informations, consultez [Désinstallation de l' AWS CLI](#) et [Installation de l' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface .

## Référence de ligne de commande pour AWS CodeBuild

AWS CLI Fournit des commandes pour l'automatisation. AWS CodeBuild Utilisez les informations de cette rubrique en complément du [guide de l'AWS Command Line Interface utilisateur](#) et de la [AWS CLI référence pour AWS CodeBuild](#).

Ce n'est pas ce que vous recherchez ? Si vous souhaitez utiliser le AWS SDKs pour appeler CodeBuild, consultez le [AWS SDKs et référence des outils](#).

Pour utiliser les informations de cette rubrique, vous devez déjà l'avoir installé AWS CLI et configuré pour une utilisation avec CodeBuild, comme décrit dans [Interagissez à CodeBuild l'aide du AWS CLI](#).

Pour utiliser le AWS CLI pour spécifier le point de terminaison pour CodeBuild, voir [Spécifiez le AWS CodeBuild point de terminaison \(AWS CLI\)](#).

Exécutez cette commande pour obtenir une liste de CodeBuild commandes.

```
aws codebuild help
```

Exécutez cette commande pour obtenir des informations sur une CodeBuild commande, dont le nom *command-name* est indiqué.

```
aws codebuild command-name help
```

CodeBuild les commandes incluent :

- `batch-delete-builds`: Supprime une ou plusieurs versions intégrées. CodeBuild Pour de plus amples informations, veuillez consulter [Suppression de générations \(AWS CLI\)](#).
- `batch-get-builds` : Obtient des informations sur plusieurs générations dans CodeBuild. Pour de plus amples informations, veuillez consulter [Affichage des détails d'une génération \(AWS CLI\)](#).
- `batch-get-projects` : Obtient des informations sur un ou plusieurs projets de génération spécifiés. Pour de plus amples informations, veuillez consulter [Affichage des détails d'un projet de génération \(AWS CLI\)](#).
- `create-project` : crée un projet de génération. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).
- `delete-project` : supprime un projet de génération Pour de plus amples informations, veuillez consulter [Suppression d'un projet de génération \(AWS CLI\)](#).
- `list-builds`: répertorie les noms des ressources Amazon (ARNs) pour les versions intégrées CodeBuild. Pour de plus amples informations, veuillez consulter [Afficher la liste des build IDs \(AWS CLI\)](#).
- `list-builds-for-project`: obtient la liste des versions IDs associées à un projet de construction spécifié. Pour de plus amples informations, veuillez consulter [Afficher la liste des versions IDs pour un projet de construction \(AWS CLI\)](#).
- `list-curated-environment-images`: Obtient une liste d'images Docker gérées CodeBuild que vous pouvez utiliser pour vos builds. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).
- `list-projects` : Obtient une liste de noms de projet de génération. Pour de plus amples informations, veuillez consulter [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- `start-build` : démarre l'exécution d'une génération. Pour de plus amples informations, veuillez consulter [Exécution d'une génération \(AWS CLI\)](#).
- `stop-build` : Tente d'arrêter l'exécution de la génération spécifiée. Pour de plus amples informations, veuillez consulter [Arrêt d'une génération \(AWS CLI\)](#).
- `update-project` : Modifie des informations sur le projet de génération spécifié. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).

# AWS SDKs et référence d'outils pour AWS CodeBuild

Pour utiliser l'un AWS SDKs des outils d'automatisation AWS CodeBuild, consultez les ressources suivantes.

Si vous souhaitez utiliser le AWS CLI pour exécuter CodeBuild, consultez le [Référence des commandes en ligne](#).

## Support AWS SDKs et outils pour AWS CodeBuild

Les outils AWS SDKs et outils suivants sont pris en charge CodeBuild :

- Le [kit SDK AWS pour C++](#). Pour plus d'informations, consultez la section [Aws : : CodeBuild](#) namespace du manuel de référence de l'AWS API SDK for C++.
- Le [kit SDK AWS pour Go](#). Pour plus d'informations, consultez la section [codebuild](#) du manuel de référence de l'AWS API SDK for Go.
- Le [kit SDK AWS pour Java](#). Pour de plus amples informations, veuillez consulter les sections `com.amazonaws.services.codebuild` et `com.amazonaws.services.codebuild.model` du document [Référence d'API du kit AWS SDK pour Java](#).
- Le [AWS SDK pour JavaScript le navigateur et le AWS SDK pour JavaScript](#) le fichier Node.js. Pour plus d'informations, consultez la [classe : AWS. CodeBuild](#) section du AWS SDK pour la référence des JavaScript API.
- Le [kit SDK AWS pour .NET](#). Pour plus d'informations, consultez les sections d'espaces de noms [Amazon.CodeBuild](#) et [Amazon.CodeBuild.Model](#) du document AWS SDK for .NET API Reference.
- Le [kit SDK AWS pour PHP](#). Pour plus d'informations, consultez la section [Namespace Aws \CodeBuild](#) du document AWS SDK for PHP API Reference.
- Le [kit SDK AWS pour Python \(Boto3\)](#). Pour plus d'informations, consultez la section [CodeBuild](#) de la documentation sur Boto 3.
- Le [kit SDK AWS pour Ruby](#). Pour plus d'informations, consultez la section [Module: Aws::CodeBuild](#) du document AWS SDK for Ruby API Reference.
- Les [AWS outils pour PowerShell](#). Pour plus d'informations, consultez la [AWS CodeBuild](#) section de référence des AWS outils pour les PowerShell applets de commande.

# Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque kit SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

| Documentation SDK                           | Exemples de code                                             |
|---------------------------------------------|--------------------------------------------------------------|
| <a href="#">AWS SDK pour C++</a>            | <a href="#">AWS SDK pour C++ exemples de code</a>            |
| <a href="#">AWS CLI</a>                     | <a href="#">AWS CLI exemples de code</a>                     |
| <a href="#">AWS SDK pour Go</a>             | <a href="#">AWS SDK pour Go exemples de code</a>             |
| <a href="#">AWS SDK pour Java</a>           | <a href="#">AWS SDK pour Java exemples de code</a>           |
| <a href="#">AWS SDK pour JavaScript</a>     | <a href="#">AWS SDK pour JavaScript exemples de code</a>     |
| <a href="#">AWS SDK pour Kotlin</a>         | <a href="#">AWS SDK pour Kotlin exemples de code</a>         |
| <a href="#">AWS SDK pour .NET</a>           | <a href="#">AWS SDK pour .NET exemples de code</a>           |
| <a href="#">AWS SDK pour PHP</a>            | <a href="#">AWS SDK pour PHP exemples de code</a>            |
| <a href="#">Outils AWS pour PowerShell</a>  | <a href="#">Outils AWS pour PowerShell exemples de code</a>  |
| <a href="#">AWS SDK pour Python (Boto3)</a> | <a href="#">AWS SDK pour Python (Boto3) exemples de code</a> |
| <a href="#">AWS SDK pour Ruby</a>           | <a href="#">AWS SDK pour Ruby exemples de code</a>           |
| <a href="#">AWS SDK pour Rust</a>           | <a href="#">AWS SDK pour Rust exemples de code</a>           |
| <a href="#">AWS SDK pour SAP ABAP</a>       | <a href="#">AWS SDK pour SAP ABAP exemples de code</a>       |
| <a href="#">AWS SDK pour Swift</a>          | <a href="#">AWS SDK pour Swift exemples de code</a>          |

Pour voir des exemples spécifiques à ce service, consultez [Exemples de code à CodeBuild utiliser AWS SDKs](#).

### Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien [Provide feedback](#) (Fournir un commentaire) en bas de cette page.

## Spécifiez le AWS CodeBuild point de terminaison

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) ou l'un des AWS SDKs pour spécifier le point de terminaison utilisé par AWS CodeBuild. Un point de terminaison CodeBuild est disponible pour chaque région. En plus d'un point de terminaison régional, quatre régions ont également un point de terminaison FIPS (Federal Information Processing Standards). Pour de plus amples informations sur les points de terminaison FIPS, veuillez consulter [Présentation de FIPS 140-2](#).

La spécification d'un point de terminaison est facultative. Si vous n'indiquez pas explicitement CodeBuild quel point de terminaison utiliser, le service utilise le point de terminaison associé à la région utilisée par votre AWS compte. CodeBuild ne prend jamais par défaut un point de terminaison FIPS. Si vous souhaitez utiliser un point de terminaison FIPS, vous devez lui associer CodeBuild via une des méthodes suivantes.

### Note

Vous pouvez utiliser un alias ou un nom de région pour spécifier un point de terminaison à l'aide d'un AWS SDK. Si vous utilisez le AWS CLI, vous devez utiliser le nom complet du point de terminaison.

Pour les points de terminaison pouvant être utilisés avec CodeBuild, voir [CodeBuild Régions et points de terminaison](#).

### Rubriques

- [Spécifiez le AWS CodeBuild point de terminaison \(AWS CLI\)](#)
- [Spécifiez le AWS CodeBuild point de terminaison \(AWS SDK\)](#)

## Spécifiez le AWS CodeBuild point de terminaison (AWS CLI)

Vous pouvez utiliser le AWS CLI pour spécifier le point de terminaison par lequel AWS CodeBuild on accède en utilisant l'`--endpoint-url` argument dans n'importe quelle CodeBuild commande. Par exemple, exécutez cette commande pour obtenir une liste des noms de version de projet à l'aide du point de terminaison FIPS (Federal Information Processing Standards) dans la région USA Est (Virginie du Nord) :

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Incluez `https://` au début du point de terminaison.

L'`--endpoint-url` AWS CLI argument est disponible pour tous les AWS services. Pour plus d'informations à ce sujet et sur d'autres AWS CLI arguments, consultez [AWS CLI la section Référence des commandes](#).

## Spécifiez le AWS CodeBuild point de terminaison (AWS SDK)

Vous pouvez utiliser un AWS SDK pour spécifier le point de terminaison par lequel AWS CodeBuild vous accédez. Bien que cet exemple utilise le [AWS SDK for Java](#), vous pouvez spécifier le point de terminaison avec l' AWS SDKsautre.

Utilisez `withEndpointConfiguration` cette méthode lors de la construction du client `AWSCodeBuild`. Voici le format à utiliser :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
 "region")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
 build();
```

Pour plus d'informations sur `AWSCodeBuildClientBuilder`, voir [Classe AWSCodeBuildClientBuilder](#).

Les informations d'identification utilisées dans `withCredentials` doivent être de type `AWSCredentialsProvider`. Pour plus d'informations, consultez la section [Utilisation des AWS informations d'identification](#).

N'incluez pas `https://` au début du point de terminaison.

Si vous souhaitez spécifier un point de terminaison non conforme à la norme FIPS, vous pouvez utiliser la région plutôt que le point de terminaison. Par exemple, pour spécifier le point de terminaison dans la région de l'est des États-Unis (Virginie du Nord), vous pouvez utiliser à la `us-east-1` place du nom complet du point de terminaison, `codebuild.us-east-1.amazonaws.com`.

Si vous souhaitez spécifier un point de terminaison FIPS, vous pouvez utiliser un alias pour simplifier votre code. Seuls les points de terminaison FIPS ont un alias. Les autres points de terminaison doivent être spécifiés à l'aide de leur nom de région ou de leur nom complet.

Le tableau suivant répertorie les alias pour chacun des quatre points de terminaison FIPS disponibles :

| Nom de la région                  | Région    | Point de terminaison                   | Alias          |
|-----------------------------------|-----------|----------------------------------------|----------------|
| USA Est<br>(Virginie du Nord)     | us-east-1 | codebuild-fips.us-east-1.amazonaws.com | us-east-1-fips |
| USA Est<br>(Ohio)                 | us-east-2 | codebuild-fips.us-east-2.amazonaws.com | us-east-2-fips |
| USA Ouest<br>(Californie du Nord) | us-west-1 | codebuild-fips.us-west-1.amazonaws.com | us-west-1-fips |
| USA Ouest<br>(Oregon)             | us-west-2 | codebuild-fips.us-west-2.amazonaws.com | us-west-2-fips |

Pour spécifier l'utilisation du point de terminaison FIPS dans la région ouest des États-Unis (Oregon) à l'aide d'un alias :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
```



```
build();
```

Pour spécifier l'utilisation du point de terminaison non FIPS dans la région de l'est des États-Unis (Virginie du Nord) :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
 "us-east-1")).
 withCredentials(new AWSSStaticCredentialsProvider(sessionCredentials)).
 build();
```

Pour spécifier l'utilisation du point de terminaison non FIPS dans la région Asie-Pacifique (Mumbai) :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
 "ap-south-1")).
 withCredentials(new AWSSStaticCredentialsProvider(sessionCredentials)).
 build();
```

## Utilisez AWS CodeBuild with AWS CodePipeline pour tester le code et exécuter des builds

Vous pouvez automatiser votre processus de publication en l'utilisant AWS CodePipeline pour tester votre code et exécuter vos builds avec AWS CodeBuild.

Le tableau suivant répertorie les tâches et les méthodes disponibles pour ces tâches. L'utilisation du AWS SDKs pour accomplir ces tâches n'entre pas dans le cadre de cette rubrique.

| Sous-tâche           | Approches disponibles                                                  | Approches décrites dans cette rubrique                                                                                                                       |
|----------------------|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Créez un pipeline de | <ul style="list-style-type: none"><li>• CodePipeline console</li></ul> | <ul style="list-style-type: none"><li>• <a href="#">Utilisation de la console CodePipeline</a></li><li>• <a href="#">Utilisation de l' AWS CLI</a></li></ul> |

| Sous-tâche                                                                                                | Approches disponibles                                                                                           | Approches décrites dans cette rubrique                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| livraison continue (CD)<br>CodePipeline qui automatise les builds avec CodeBuild                          | <ul style="list-style-type: none"> <li>• AWS CLI</li> <li>• AWS SDKs</li> </ul>                                 | <ul style="list-style-type: none"> <li>• Vous pouvez adapter les informations de cette rubrique pour utiliser le AWS SDKs. Pour plus d'informations, consultez la documentation relative aux <code>create-pipeline</code> actions correspondant à votre langage de programmation dans la <a href="#">SDKs</a> section Outils pour Amazon Web Services ou <a href="#">CreatePipeline</a> dans le Guide de référence des AWS CodePipeline API.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Ajoutez l'automatisation des tests et du CodeBuild développement à un pipeline existant dans CodePipeline | <ul style="list-style-type: none"> <li>• CodePipeline console</li> <li>• AWS CLI</li> <li>• AWS SDKs</li> </ul> | <ul style="list-style-type: none"> <li>• <a href="#">Utiliser la console CodePipeline pour automatiser les générations</a></li> <li>• <a href="#">Utiliser la console CodePipeline pour automatiser les tests</a></li> <li>• Pour le AWS CLI, vous pouvez adapter les informations de cette rubrique afin de créer un pipeline contenant une action de CodeBuild création ou une action de test. Pour plus d'informations, voir <a href="#">Modifier un pipeline (AWS CLI)</a> et la <a href="#">référence de structure du CodePipeline pipeline</a> dans le Guide de AWS CodePipeline l'utilisateur.</li> <li>• Vous pouvez adapter les informations de cette rubrique pour utiliser le AWS SDKs. Pour plus d'informations, consultez la documentation relative aux <code>update-pipeline</code> actions correspondant à votre langage de programmation dans la <a href="#">SDKs</a> section Outils pour Amazon Web Services ou <a href="#">UpdatePipeline</a> dans le Guide de référence des AWS CodePipeline API.</li> </ul> |

## Rubriques

- [Conditions préalables](#)
- [Créez un pipeline qui utilise CodeBuild \(CodePipeline console\)](#)
- [Créez un pipeline qui utilise CodeBuild \(AWS CLI\)](#)
- [Ajouter une CodeBuild action de génération à un pipeline \(CodePipeline console\)](#)

- [Ajouter une action de CodeBuild test à un pipeline \(CodePipeline console\)](#)

## Conditions préalables

1. Répondez aux questions de [Planification d'une génération](#).
2. Si vous utilisez un utilisateur pour accéder CodePipeline au lieu d'un compte AWS root ou d'un utilisateur administrateur, associez la politique gérée nommée `AWSCodePipelineFullAccess` à l'utilisateur (ou au groupe IAM auquel appartient l'utilisateur). L'utilisation d'un compte AWS root n'est pas recommandée. Cette stratégie accorde à l'utilisateur l'autorisation de créer le pipeline dans CodePipeline. Pour plus d'informations, consultez la section [Joindre des politiques gérées](#) dans le guide de l'utilisateur.

### Note

L'entité IAM qui attache la politique à l'utilisateur (ou au groupe IAM auquel appartient l'utilisateur) doit être autorisée dans IAM à attacher des politiques. Pour plus d'informations, consultez la section [Délégation d'autorisations pour administrer les utilisateurs, les groupes et les informations d'identification IAM](#) dans le guide de l'utilisateur.

3. Créez un rôle de CodePipeline service, si aucun rôle n'est déjà disponible dans votre AWS compte. CodePipeline utilise ce rôle de service pour interagir avec d'autres AWS services AWS CodeBuild, y compris en votre nom. Par exemple, pour utiliser le AWS CLI pour créer un rôle CodePipeline de service, exécutez la `create-role` commande IAM :

Pour Linux, macOS ou Unix :

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-role-policy-document '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service": "codepipeline.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'
```

Pour Windows :

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-role-policy-document "{\"Version\":\"2012-10-17\", \"Statement\": [{\"Effect
```

```
\":\\"Allow\\",\\"Principal\\":{\\"Service\\":\\"codepipeline.amazonaws.com\\"},\\"Action\\":\\"sts:AssumeRole\\"}]}"
```

#### Note

L'entité IAM qui crée ce rôle de CodePipeline service doit être autorisée dans IAM à créer des rôles de service.

- Après avoir créé un rôle de CodePipeline service ou identifié un rôle existant, vous devez ajouter la politique de rôle de CodePipeline service par défaut au rôle de service, comme décrit dans la section [Révision de la politique de rôle de CodePipeline service par défaut](#) dans le Guide de AWS CodePipeline l'utilisateur, si elle ne fait pas déjà partie de la politique du rôle.

#### Note

L'entité IAM qui ajoute cette politique de rôle de CodePipeline service doit être autorisée dans IAM à ajouter des politiques de rôle de service aux rôles de service.

- Créez et chargez le code source dans un type de référentiel pris en charge par CodeBuild et CodePipeline CodeCommit, tel qu'Amazon S3, Bitbucket ou GitHub. Le code source doit contenir un fichier de spécification de génération, mais vous pourrez en déclarer un lorsque vous définirez un projet de génération ultérieurement dans cette rubrique. Pour de plus amples informations, veuillez consulter [Référence des spécifications de génération](#).

#### Important

Si vous prévoyez d'utiliser le pipeline pour déployer le code source généré, l'artefact de sortie de génération doit être compatible avec le système de déploiement que vous utilisez.

- Pour OpsWorks, voir [Source de l'application](#) et [Utilisation CodePipeline avec OpsWorks](#) dans le guide de OpsWorks l'utilisateur.

## Créez un pipeline qui utilise CodeBuild (CodePipeline console)

Utilisez la procédure suivante pour créer un pipeline qui permet CodeBuild de créer et de déployer votre code source.

Pour créer un pipeline qui teste uniquement votre code source :

- Utilisez la procédure suivante pour créer le pipeline, puis supprimez les étapes de génération et bêta du pipeline. Utilisez ensuite la procédure [Ajouter une action de CodeBuild test à un pipeline \(CodePipeline console\)](#) décrite dans cette rubrique pour ajouter au pipeline une action de test qui utilise CodeBuild.
- Utilisez l'une des autres procédures de cette rubrique pour créer le pipeline, puis utilisez la procédure [Ajouter une action de CodeBuild test à un pipeline \(CodePipeline console\)](#) décrite dans cette rubrique pour ajouter au pipeline une action de test qui utilise CodeBuild.

Pour utiliser l'assistant de création de pipeline CodePipeline pour créer un pipeline qui utilise CodeBuild

1. Connectez-vous au AWS Management Console en utilisant :

- Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le guide de l'utilisateur.
- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
```

```
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Ouvrez la AWS CodePipeline console à l'adresse <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Dans le sélecteur de AWS région, choisissez la AWS région dans laquelle se trouvent les AWS ressources de votre projet de construction. Il doit s'agir d'une AWS région prise CodeBuild en charge. Pour plus d'informations, consultez [AWS CodeBuild](#) dans le Référence générale d'Amazon Web Services.
4. Créez un pipeline. Si une page CodePipeline d'informations s'affiche, choisissez Create pipeline. Si une page Tous les pipelines s'affiche, choisissez Créer un pipeline.
5. Sur la page Étape 1 : Choisir des paramètres de pipeline, pour Nom du pipeline, saisissez un nom pour le pipeline, par exemple, **CodeBuildDemoPipeline**. Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.
6. Pour Nom du rôle, sélectionnez l'une des options suivantes :

Choisissez New service role (Nouveau rôle de service) et dans Nom du rôle, saisissez le nom de votre nouveau rôle de service.

Choisissez Existing service role (Rôle de service existant), puis choisissez le rôle de service CodePipeline que vous avez créé ou identifié dans le cadre des prérequis de cette rubrique.

7. Pour Artifact store (Magasin d'artefacts), effectuez l'une des opérations suivantes :
  - Choisissez Emplacement par défaut pour utiliser le magasin d'artefacts par défaut, tel que le compartiment d'artefacts S3 désigné par défaut, pour votre pipeline dans la AWS région que vous avez sélectionnée pour votre pipeline.
  - Choisissez Emplacement personnalisé si vous avez déjà créé un magasin d'artefacts, tel qu'un bucket d'artefacts S3, dans la même AWS région que votre pipeline.

#### Note

Il ne s'agit pas du compartiment source du code source pour votre pipeline. Il s'agit du magasin d'artefacts pour votre pipeline. Un magasin d'artefacts distinct, tel qu'un

compartiment S3, est requis pour chaque pipeline, dans la même AWS région que le pipeline.

8. Choisissez Suivant.
9. Sur la page Étape 2 : Ajouter une étape source, pour Fournisseur de source, effectuez l'une des actions suivantes :
  - Si votre code source est stocké dans un compartiment S3, choisissez Amazon S3. Pour Bucket (Compartiment), sélectionnez le compartiment S3 qui contient votre code source. Pour S3 object key (Clé d'objet S3), saisissez le nom du fichier qui contient le code source (par exemple *file-name*.zip). Choisissez Suivant.
  - Si votre code source est stocké dans un AWS CodeCommit dépôt, choisissez CodeCommit. Pour Repository name, choisissez le nom du référentiel qui contient le code source. Pour Branch name (Nom de branche), choisissez le nom de la branche qui représente la version du code source à générer. Choisissez Suivant.
  - Si votre code source est stocké dans un GitHub dépôt, choisissez GitHub. Choisissez Connect to GitHub, puis suivez les instructions pour vous authentifier auprès GitHub de. Pour Repository, choisissez le nom du référentiel qui contient le code source. Pour Branch (Branche), choisissez le nom de la branche qui représente la version du code source à générer.

Choisissez Suivant.

10. Sur la page Étape 3: Ajouter une étape de génération, pour Fournisseur de génération, choisissez CodeBuild.
11. Si vous avez déjà un projet de construction que vous souhaitez utiliser, dans Nom du projet, choisissez le nom du projet de construction et passez à l'étape suivante de cette procédure.

Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

Si vous choisissez un projet de construction existant, les paramètres d'artefact de sortie de construction doivent déjà être définis (même s'ils les CodePipeline remplacent). Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).

**⚠ Important**

Si vous activez les webhooks pour un CodeBuild projet et que le projet est utilisé comme étape de construction CodePipeline, deux versions identiques sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la console AWS CodeBuild, décochez la case Webhook. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).

12. Sur la page Étape 4 : Ajouter une étape de déploiement, effectuez l'une des opérations suivantes :

- Si vous ne souhaitez pas déployer l'artefact de sortie de génération, choisissez Ignorer et confirmez ce choix lorsque vous y êtes invité.
- Si vous souhaitez déployer l'artefact de sortie de génération, pour Deploy provider (Fournisseur de déploiement), choisissez un fournisseur de déploiement, puis spécifiez les paramètres lorsque vous y êtes invité.

Choisissez Suivant.

13. Sur la page Vérification, vérifiez vos sélections, puis choisissez Créer un pipeline.
14. Une fois que le pipeline s'exécute correctement, vous pouvez obtenir l'artefact de sortie de la génération. Le pipeline étant affiché dans la CodePipeline console, dans l'action Construire, choisissez l'infobulle. Notez la valeur de l'artefact de sortie (par exemple, MyAppBuild).

**i Note**

Vous pouvez également obtenir l'artefact de sortie du build en cliquant sur le lien Build artefacts sur la page des détails du build dans la CodeBuild console. Pour accéder à cette page, ignorez les autres étapes de cette procédure et consultez [Affichage des détails d'une génération \(console\)](#).

15. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
16. Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format codepipeline-*region-ID-random-number*. Vous



pouvez utiliser le AWS CLI pour exécuter la CodePipeline `get-pipeline` commande afin d'obtenir le nom du bucket, où se *my-pipeline-name* trouve le nom d'affichage de votre pipeline :

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dans la sortie, l'objet `pipeline` contient un objet `artifactStore` contenant une valeur `location` avec le nom du compartiment.

17. Ouvrez le dossier correspondant au nom de votre pipeline (selon la longueur du nom du pipeline, le nom de dossier peut être tronqué), puis ouvrez le dossier correspondant à la valeur pour Output artifact (Artefact de sortie) que vous avez notée précédemment.
18. Extrayez le contenu du fichier . Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage Last Modified. (Vous pouvez attribuer au fichier l'extension .zip pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.
19. Si vous avez demandé CodePipeline de déployer l'artefact de sortie de génération, utilisez les instructions du fournisseur de déploiement pour accéder à l'artefact de sortie de génération sur les cibles de déploiement.

## Créez un pipeline qui utilise CodeBuild (AWS CLI)

Utilisez la procédure suivante pour créer un pipeline qui servira CodeBuild à générer votre code source.

**AWS CLI** Pour créer un pipeline qui déploie votre code source créé ou qui teste uniquement votre code source, vous pouvez adapter les instructions de la section [Modifier un pipeline \(AWS CLI\)](#) et la [référence de structure du CodePipeline pipeline](#) dans le guide de l'AWS CodePipeline utilisateur.

1. Créez ou identifiez un projet de construction dans CodeBuild. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération](#).

### Important

Le projet de génération doit définir des paramètres d'artefact de sortie de génération (même si CodePipeline va remplacer). Pour de plus amples informations, consultez la description de `artifacts` dans [Création d'un projet de génération \(AWS CLI\)](#).

2. Assurez-vous d'avoir configuré la clé AWS CLI d' AWS accès et la clé d'accès AWS secrète correspondant à l'une des entités IAM décrites dans cette rubrique. Pour de plus amples informations, veuillez consulter [Préparation de l' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface .
3. Créez un fichier au format JSON représentant la structure du pipeline. Nommez le fichier `create-pipeline.json` ou avec un nom similaire. Par exemple, cette structure au format JSON crée un pipeline avec une action source qui référence un compartiment d'entrée S3 et une action de génération utilisant CodeBuild :

```
{
 "pipeline": {
 "roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-name>",
 "stages": [
 {
 "name": "Source",
 "actions": [
 {
 "inputArtifacts": [],
 "name": "Source",
 "actionTypeId": {
 "category": "Source",
 "owner": "AWS",
 "version": "1",
 "provider": "S3"
 },
 "outputArtifacts": [
 {
 "name": "MyApp"
 }
],
 "configuration": {
 "S3Bucket": "<bucket-name>",
 "S3ObjectKey": "<source-code-file-name.zip>"
 },
 "runOrder": 1
 }
]
 },
 {
 "name": "Build",
 "actions": [
```

```
{
 "inputArtifacts": [
 {
 "name": "MyApp"
 }
],
 "name": "Build",
 "actionTypeId": {
 "category": "Build",
 "owner": "AWS",
 "version": "1",
 "provider": "CodeBuild"
 },
 "outputArtifacts": [
 {
 "name": "default"
 }
],
 "configuration": {
 "ProjectName": "<build-project-name>"
 },
 "runOrder": 1
}
]
}
],
"artifactStore": {
 "type": "S3",
 "location": "<CodePipeline-internal-bucket-name>"
},
"name": "<my-pipeline-name>",
"version": 1
}
}
```

Dans ces données au format JSON :

- La valeur de `roleArn` doit correspondre à l'ARN du rôle de CodePipeline service que vous avez créé ou identifié dans le cadre des prérequis.
- Les valeurs de `S3Bucket` et `S3ObjectKey` dans `configuration` supposent que le code source est stocké dans un compartiment S3. Pour les paramètres pour d'autres types

de référentiel de code source, veuillez consulter [Référence sur la structure du pipeline CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline .

- La valeur de `ProjectName` est le nom du projet de CodeBuild construction que vous avez créé plus tôt dans cette procédure.
- La valeur de `location` est le nom du compartiment S3 utilisé par ce pipeline. Pour de plus amples informations, veuillez consulter [Création d'une stratégie pour un compartiment S3 en vue de l'utiliser comme magasin d'artefacts pour CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline .
- La valeur de `name` est le nom de ce pipeline. Tous les noms de pipeline doivent être uniques dans votre compte.

Bien que ces données ne décrivent qu'une action source et une action de génération, vous pouvez ajouter des actions pour les activités liées aux tests, au déploiement de l'artefact de sortie de génération, à l'invocation de AWS Lambda fonctions, etc. Pour de plus amples informations, veuillez consulter [Référence sur la structure du pipeline AWS CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline .

4. Accédez au dossier contenant le fichier JSON, puis exécutez la CodePipeline [create-pipeline](#) commande en spécifiant le nom du fichier :

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

#### Note

Vous devez créer le pipeline dans une AWS région où il CodeBuild est pris en charge. Pour plus d'informations, consultez [AWS CodeBuild](#) dans le Référence générale d'Amazon Web Services.

Les données au format JSON apparaissent dans la sortie et CodePipeline crée le pipeline.

5. Pour obtenir des informations sur l'état du pipeline, exécutez la CodePipeline [get-pipeline-state](#) commande en spécifiant le nom du pipeline :

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Dans la sortie, recherchez les informations qui confirment que la génération a réussi. Des ellipses (...) sont utilisées pour les données qui ont été omises par souci de concision.

```
{
 ...
 "stageStates": [
 ...
 {
 "actionStates": [
 {
 "actionName": "CodeBuild",
 "latestExecution": {
 "status": "SUCCEEDED",
 ...
 },
 ...
 },
 ...
]
 }
]
}
```

Si vous exécutez cette commande trop tôt, vous risquez de ne voir aucune information sur l'action de génération. Vous devrez peut-être exécuter cette commande plusieurs fois jusqu'à ce que le pipeline ait fini d'exécuter l'action de génération.

6. Après une génération réussie, suivez ces instructions pour obtenir l'artefact de sortie de la génération. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.

#### Note

Vous pouvez également obtenir l'artefact de sortie de la génération en choisissant le lien Artefacts de génération sur la page des détails de génération associées dans la console CodeBuild. Pour accéder à cette page, ignorez les autres étapes de cette procédure et consultez [Affichage des détails d'une génération \(console\)](#).

7. Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format `codepipeline-<region-ID>-<random-number>`. Vous pouvez obtenir le nom du compartiment à partir du `create-pipeline.json` fichier ou exécuter la `CodePipeline get-pipeline` commande pour obtenir le nom du compartiment.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

Dans la sortie, l'objet pipeline contient un objet artifactStore contenant une valeur location avec le nom du compartiment.

8. Ouvrez le dossier correspondant au nom de votre pipeline (par exemple, <pipeline-name>).
9. Dans ce dossier, ouvrez le dossier nommé default (par défaut).
10. Extrayez le contenu du fichier . Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage Last Modified. (Vous pouvez attribuer au fichier une extension .zip pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.

## Ajouter une CodeBuild action de génération à un pipeline (CodePipeline console)

1. Connectez-vous au AWS Management Console en utilisant :
  - Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le guide de l'utilisateur.
  - Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
  - Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
```

```
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Ouvrez la CodePipeline console à l'adresse <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Dans le sélecteur de AWS région, choisissez la AWS région où se trouve votre pipeline. Il doit s'agir d'une région prise CodeBuild en charge. Pour plus d'informations, consultez [CodeBuild](#) dans le Référence générale d'Amazon Web Services.
4. Sur la page Tous les pipelines, choisissez le nom du pipeline.
5. Sur la page des détails du pipeline, dans l'action Source, sélectionnez l'info-bulle. Notez la valeur de l'artefact de sortie (par exemple, MyApp).

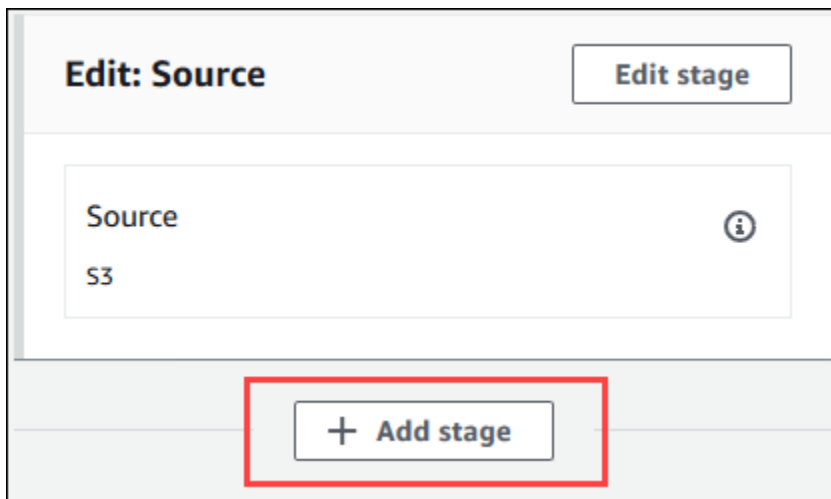
 **Note**

Cette procédure vous explique comment ajouter une action de génération lors d'une étape de génération, entre les étapes Source et Bêta. Si vous souhaitez ajouter l'action de génération ailleurs, sélectionnez l'info-bulle de l'action juste avant l'emplacement où vous voulez ajouter l'action de génération, et notez la valeur du champ Artefact de sortie.

6. Choisissez Modifier.
7. Entre les étapes Source et Bêta, choisissez Ajouter une étape.

 **Note**

Cette procédure vous explique comment ajouter une étape de génération entre les étapes Source et Bêta du pipeline. Pour ajouter une action de génération à une étape existante, choisissez Edit stage (Modifier une étape) à l'étape, puis passez directement à l'étape 8 de cette procédure. Pour ajouter ailleurs l'étape de génération, choisissez Add stage (Ajouter une étape) à l'emplacement voulu.



8. Pour Stage name (Nom de l'étape), saisissez le nom de l'étape de génération (par exemple, **Build**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
9. Au sein de l'étape sélectionnée, choisissez Ajouter une action.

**Note**

Cette procédure vous explique comment ajouter l'action de génération lors d'une étape de génération. Pour ajouter ailleurs l'action de génération, choisissez Ajouter une action à l'emplacement voulu. Vous devrez peut-être d'abord choisir l'étape Modifier dans l'étape existante dans laquelle vous voulez générer l'action.

10. Dans Modifier l'action, pour Nom de l'action, saisissez un nom pour l'action (par exemple, **CodeBuild**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
11. Pour Action provider (Fournisseur d'action), choisissez CodeBuild.
12. Si vous avez déjà un projet de construction que vous souhaitez utiliser, dans Nom du projet, choisissez le nom du projet de construction et passez à l'étape suivante de cette procédure.

Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

Si vous choisissez un projet de construction existant, les paramètres d'artefact de sortie de construction doivent déjà être définis (même s'ils les CodePipeline remplacent). Pour de plus amples informations, consultez la description d'Artefacts dans [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).



 Important

Si vous activez les webhooks pour un CodeBuild projet et que le projet est utilisé comme étape de construction CodePipeline, deux versions identiques sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la CodeBuild console, décochez la case Webhook. Pour de plus amples informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).

13. Pour Input artifacts (Artefacts d'entrée), sélectionnez l'artefact de sortie que vous avez notée précédemment dans cette procédure.
14. Pour Artefacts de sortie, saisissez un nom pour l'artefact de sortie (par exemple, **MyAppBuild**).
15. Choisissez Add action.
16. Choisissez Save (Enregistrer), puis Save (Enregistrer) pour enregistrer les modifications apportées au pipeline.
17. Choisissez Release Change.
18. Une fois que le pipeline s'exécute correctement, vous pouvez obtenir l'artefact de sortie de la génération. Le pipeline étant affiché dans la CodePipeline console, dans l'action Construire, choisissez l'infobulle. Notez la valeur de l'artefact de sortie (par exemple, MyAppBuild).

 Note

Vous pouvez également obtenir l'artefact de sortie du build en cliquant sur le lien Build artefacts sur la page des détails du build dans la CodeBuild console. Pour accéder à cette page, consultez [Affichage des détails d'une génération \(console\)](#), puis passez directement à l'étape 31 de cette procédure.

19. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
20. Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format codepipeline-*region-ID-random-number*. Vous pouvez utiliser le AWS CLI pour exécuter la CodePipeline get-pipeline commande afin d'obtenir le nom du bucket :

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dans la sortie, l'objet `pipeline` contient un objet `artifactStore` contenant une valeur `location` avec le nom du compartiment.

21. Ouvrez le dossier correspondant au nom de votre pipeline (selon la longueur du nom du pipeline, le nom de dossier peut être tronqué), puis ouvrez le dossier correspondant à la valeur pour Output artifact (Artefact de sortie) que vous avez notée précédemment dans cette procédure.
22. Extrayez le contenu du fichier . Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage Last Modified. (Vous pouvez attribuer au fichier l'extension `.zip` pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.
23. Si vous avez demandé CodePipeline de déployer l'artefact de sortie de génération, utilisez les instructions du fournisseur de déploiement pour accéder à l'artefact de sortie de génération sur les cibles de déploiement.

## Ajouter une action de CodeBuild test à un pipeline (CodePipeline console)

1. Connectez-vous au AWS Management Console en utilisant :
  - Votre compte AWS root. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le guide de l'utilisateur.
  - Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
  - Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
```

```
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Ouvrez la CodePipeline console sur <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Dans le sélecteur de AWS région, choisissez la AWS région où se trouve votre pipeline. Il doit s'agir d'une AWS région prise CodeBuild en charge. Pour plus d'informations, consultez [AWS CodeBuild](#) dans le Référence générale d'Amazon Web Services.
4. Sur la page Tous les pipelines, choisissez le nom du pipeline.
5. Sur la page des détails du pipeline, dans l'action Source, sélectionnez l'info-bulle. Notez la valeur de l'artefact de sortie (par exemple, MyApp).

 Note

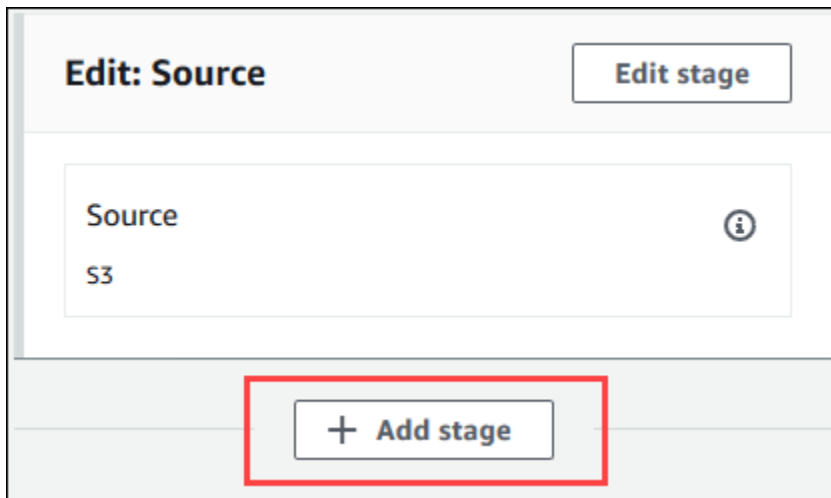
Cette procédure vous explique comment ajouter une action de test lors d'une étape de test, entre les étapes Source et Bêta. Si vous souhaitez ajouter l'action de test ailleurs, placez le pointeur de la souris sur l'action juste avant et notez la valeur pour Artefact de sortie.

6. Choisissez Modifier.
7. Immédiatement après l'étape Source, choisissez Add stage (Ajouter une étape).

 Note

Cette procédure vous explique comment ajouter au pipeline une étape de test immédiatement après l'étape Source. Pour ajouter une action de test à une étape existante, choisissez Edit stage (Modifier une étape) à l'étape, puis passez directement

à l'étape 8 de cette procédure. Pour ajouter ailleurs l'étape de test, choisissez Add stage (Ajouter une étape) à l'emplacement voulu.



8. Pour Stage name (Nom de l'étape), tapez le nom de l'étape de test (par exemple, **Test**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
9. À l'étape sélectionnez, choisissez Ajouter une action.

**Note**

Cette procédure vous explique comment ajouter l'action de test à une étape de test. Pour ajouter l'action de test ailleurs, choisissez Add action (Ajouter une action) à l'endroit voulu. Vous devrez peut-être d'abord choisir Edit (Modifier) dans l'étape existante à laquelle vous voulez ajouter l'action de test.

10. Dans Modifier l'action, pour Nom de l'action, saisissez un nom pour l'action (par exemple, **Test**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
11. Pour Action provider (Fournisseur d'actions), sous Test, choisissez CodeBuild.
12. Si vous avez déjà un projet de construction que vous souhaitez utiliser, dans Nom du projet, choisissez le nom du projet de construction et passez à l'étape suivante de cette procédure.

Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

**⚠ Important**

Si vous activez les webhooks pour un CodeBuild projet et que le projet est utilisé comme étape de construction CodePipeline, deux versions identiques sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la CodeBuild console, décochez la case Webhook. Pour de plus amples informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).

13. Pour Input artifacts (Artefacts d'entrée), sélectionnez la valeur Output artifact (Artefact de sortie) que vous avez notée précédemment dans cette procédure.
14. (Facultatif) Si vous voulez que votre action de test génère un artefact de sortie et que vous avez configuré votre spécification de génération en conséquence, pour Artefact de sortie, entrez la valeur que vous voulez affecter à l'artefact de sortie.
15. Choisissez Enregistrer.
16. Choisissez Release Change.
17. Une fois que le pipeline s'est exécuté avec succès, vous pouvez obtenir les résultats de test. Dans la phase de test du pipeline, choisissez le CodeBuild lien hypertexte pour ouvrir la page du projet de construction correspondante dans la CodeBuild console.
18. Sur la page de projet de génération, dans la zone Historique de génération, choisissez le lien hypertexte Exécution de génération.
19. Sur la page d'exécution de la compilation, dans Build logs, cliquez sur le lien hypertexte Afficher le journal complet pour ouvrir le journal de build dans la CloudWatch console Amazon.
20. Faites défiler le journal de génération pour consulter les résultats de test.

## Utilisation AWS CodeBuild avec Codecov

Codecov est un outil qui mesure la couverture de test de votre code. Codecov identifie quelles méthodes et instructions de votre code ne sont pas testées. Utilisez les résultats pour déterminer à quel niveau il pourrait être utile de créer des tests pour améliorer la qualité de votre code. Codecov est disponible pour trois des référentiels sources pris en charge par CodeBuild : GitHub Enterprise

Server GitHub et Bitbucket. Si votre projet de build utilise GitHub Enterprise Server, vous devez utiliser Codecov Enterprise.

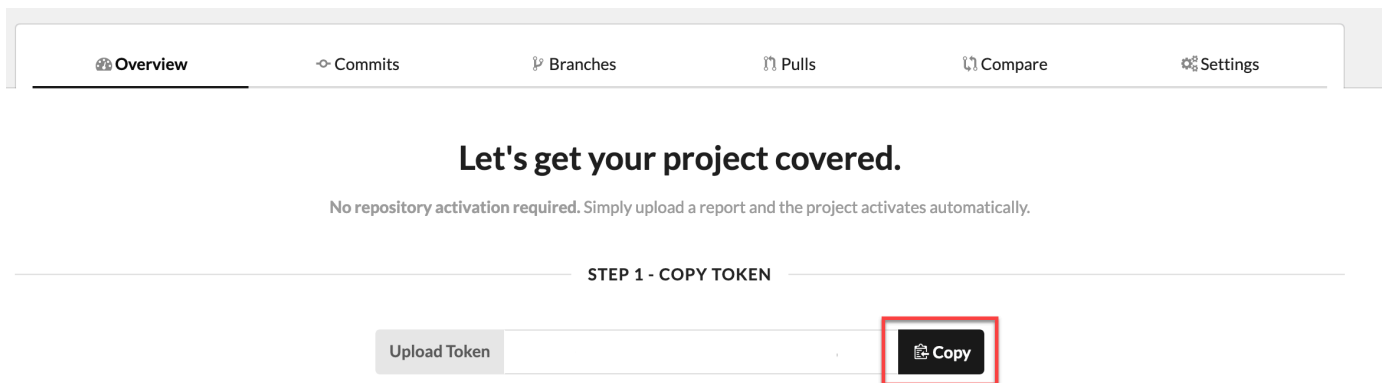
Lorsque vous exécutez la version d'un CodeBuild projet intégré à Codecov, les rapports Codecov qui analysent le code de votre référentiel sont téléchargés vers Codecov. Les journaux de génération comprennent un lien vers ces rapports. Cet exemple montre comment intégrer un projet de génération Python et Java avec Codecov. Pour obtenir la liste des langues prises en charge par Codecov, veuillez consulter [Langues prises en charge par Codecov](#) sur le site web de Codecov.

## Intégration de Codecov dans un projet de génération

Utilisez la procédure suivante pour intégrer Codecov dans un projet de construction.

Pour intégrer Codecov dans votre projet de génération

1. Accédez à <https://codecov.io/signup> et inscrivez-vous à un dépôt source GitHub ou à un dépôt de données Bitbucket. Si vous utilisez GitHub Enterprise, consultez [Codecov Enterprise](#) sur le site Web de Codecov.
2. Dans Codecov, ajoutez le référentiel pour lequel vous souhaitez couvrir le code.
3. Lorsque les informations de jeton sont affichées, choisissez Copy (Copier).



4. Ajoutez le jeton copié en tant que variable d'environnement CODECOV\_TOKEN à votre projet de génération. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).
5. Créez un fichier texte nommé `my_script.sh` dans votre référentiel. Copiez ce qui suit dans ce fichier :

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

6. Choisissez l'onglet Python ou Java selon les utilisations de votre projet de génération, puis procédez comme suit.

## Java

1. Ajoutez le JaCoCo plugin suivant pom.xml dans votre dépôt.

```
<build>
 <plugins>
 <plugin>
 <groupId>org.jacoco</groupId>
 <artifactId>jacoco-maven-plugin</artifactId>
 <version>0.8.2</version>
 <executions>
 <execution>
 <goals>
 <goal>prepare-agent</goal>
 </goals>
 </execution>
 <execution>
 <id>report</id>
 <phase>test</phase>
 <goals>
 <goal>report</goal>
 </goals>
 </execution>
 </executions>
 </plugin>
 </plugins>
</build>
```

2. Entrez les commandes suivantes dans le fichier buildspec. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

```
build:
 - mvn test -f pom.xml -fn
postbuild:
 - echo 'Connect to CodeCov'
 - bash my_script.sh
```

## Python

Entrez les commandes suivantes dans le fichier buildspec. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

```
build:
 - pip install coverage
 - coverage run -m unittest discover
postbuild:
 - echo 'Connect to CodeCov'
 - bash my_script.sh
```

- Exécutez une version du projet de génération. Un lien vers les rapports Codecov générés pour votre projet apparaît dans les journaux de génération. Utilisez ce lien pour afficher les rapports Codecov. Pour plus d'informations, consultez [Exécuter AWS CodeBuild les builds manuellement](#) et [Enregistrez les appels AWS CodeBuild d'API avec AWS CloudTrail](#). Les informations Codecov dans les journaux de génération ressemblent à ce qui suit :

```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh
```

```

 ____ _
 / ____| | | |

	/ __ \			
	__ \			

_	___	_	_	_

```

Bash-20200303-bc4d7e6

```
.[0;90m==>.[0m AWS CodeBuild detected.
```

```
... The full list of Codecov log entries has been omitted for brevity ...
```

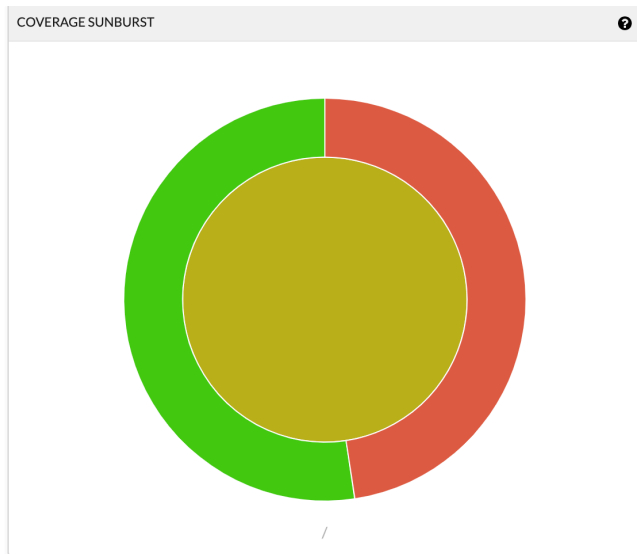
```
.
```

```
.[0;32m->.[0m View reports at .[0;36mhttps://codecov.io/github/user/test_py/commit/commit-id.[0m
```

```
[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

Les rapports se présentent comme suit :





| Files                    |    |    |   |   | Coverage |
|--------------------------|----|----|---|---|----------|
| <code>code.py</code>     | 10 | 7  | 0 | 3 | 70.00%   |
| <code>tests.py</code>    | 11 | 11 | 0 | 0 | 100.00%  |
| Project Totals (2 files) | 21 | 18 | 0 | 3 | 85.71%   |

## À utiliser AWS CodeBuild avec Jenkins

Vous pouvez utiliser le plugin Jenkins pour l'intégrer AWS CodeBuild à vos CodeBuild tâches de génération Jenkins. Au lieu d'envoyer les tâches de génération aux nœuds de génération Jenkins, vous utilisez le plug-in pour envoyer vos tâches de génération à CodeBuild. Vous n'avez ainsi plus besoin d'allouer, de configurer et de gérer des nœuds de génération Jenkins.

### Rubriques

- [Configurer Jenkins](#)
- [Installer le plug-in](#)
- [Utiliser le plugin](#)

## Configurer Jenkins

Pour plus d'informations sur la configuration de Jenkins avec le AWS CodeBuild plugin et pour télécharger le code source du plugin, consultez <https://github.com/awslabs/aws-codebuild-jenkins-plugin>.

## Installer le plug-in

Si Jenkins est déjà configuré et que vous souhaitez uniquement installer le plug-in AWS CodeBuild , accédez à votre instance Jenkins, puis dans le gestionnaire de plug-in, recherchez **CodeBuild Plugin for Jenkins**.

## Utiliser le plugin

À utiliser AWS CodeBuild avec des sources extérieures à un VPC

1. Créez un projet dans la CodeBuild console. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).
  - Choisissez la AWS région dans laquelle vous souhaitez exécuter le build.
  - (Facultatif) Définissez la configuration Amazon VPC pour permettre au conteneur de CodeBuild build d'accéder aux ressources de votre VPC.
  - Notez le nom de votre projet. Vous en aurez besoin à l'étape 3.
  - (Facultatif) Si votre référentiel source n'est pas pris en charge de manière native par CodeBuild, vous pouvez définir Amazon S3 comme type de source d'entrée pour votre projet.
2. Dans le IAMconsole, créez un utilisateur à utiliser par le plugin Jenkins.
  - Lorsque vous créez des informations d'identification pour l'utilisateur, choisissez Programmatic Access (Accès par programmation).
  - Créez une stratégie similaire à ce qui suit, puis attachez la stratégie à votre utilisateur.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Resource": [
 "arn:aws:logs:us-east-1:111122223333:log-group:/aws/codebuild/{{projectName}}:*"
],
 "Action": [
 "logs:GetLogEvents"
]
 }
],
}
```

```

 {
 "Effect": "Allow",
 "Resource": [
 "arn:aws:s3:::{{inputBucket}}"
],
 "Action": [
 "s3:GetBucketVersioning"
]
 },
 {
 "Effect": "Allow",
 "Resource": [
 "arn:aws:s3:::{{inputBucket}}/{{inputObject}}"
],
 "Action": [
 "s3:PutObject"
]
 },
 {
 "Effect": "Allow",
 "Resource": [
 "arn:aws:s3:::{{outputBucket}}/*"
],
 "Action": [
 "s3:GetObject"
]
 },
 {
 "Effect": "Allow",
 "Resource": [
 "arn:aws:codebuild:us-east-1:111122223333:project/
{{projectName}}"
],
 "Action": [
 "codebuild:StartBuild",
 "codebuild:BatchGetBuilds",
 "codebuild:BatchGetProjects"
]
 }
]
}

```

### 3. Créez un projet freestyle dans Jenkins.

- Sur la page Configure (Configurer), choisissez Add build step (Ajouter une étape de génération), puis choisissez Run build on CodeBuild (Exécuter la génération sur &ACB;).
  - Configurez votre étape de génération.
    - Fournissez des valeurs pour Region (Région), Credentials (Informations d'identification) et Project Name (Nom du projet).
    - Choisissez Use Project source (Utiliser la source de projet).
    - Enregistrez la configuration et exécutez une génération à partir de Jenkins.
4. Pour Source Code Management (Gestion de code source), choisissez la façon dont vous souhaitez récupérer votre source. Vous devrez peut-être installer le GitHub plugin (ou le plugin Jenkins pour votre fournisseur de référentiel source) sur votre serveur Jenkins.
- Sur la page Configure (Configurer), choisissez Add build step (Ajouter une étape de génération), puis choisissez Run build on AWS CodeBuild(Exécuter la génération sur &ACB;).
  - Configurez votre étape de génération.
    - Fournissez des valeurs pour Region (Région), Credentials (Informations d'identification) et Project Name (Nom du projet).
    - Choisissez Use Jenkins source (Utiliser la source Jenkins).
    - Enregistrez la configuration et exécutez une génération à partir de Jenkins.

Pour utiliser le AWS CodeBuild plugin avec le plugin Jenkins Pipeline

- Sur la page de votre projet de pipeline Jenkins, utilisez le générateur d'extraits pour générer un script de pipeline qui s'ajoute en CodeBuild tant qu'étape à votre pipeline. Cela doit générer un script semblable à ce qui suit :

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',
sourceControlType: 'jenkins'
```

## Utilisation AWS CodeBuild avec des applications sans serveur

The AWS Serverless Application Model (AWS SAM) est un framework open source permettant de créer des applications sans serveur. Pour plus d'informations, consultez le référentiel de [modèles d'applications AWS sans serveur](#) sur GitHub.

Vous pouvez l'utiliser AWS CodeBuild pour emballer et déployer des applications sans serveur conformes à la AWS SAM norme. Pour l'étape de déploiement, CodeBuild vous pouvez utiliser AWS CloudFormation. Pour automatiser la création et le déploiement d'applications sans serveur avec CodeBuild et CloudFormation, vous pouvez utiliser AWS CodePipeline.

Pour plus d'informations, consultez la section [Déploiement d'applications sans serveur](#) dans le guide du AWS Serverless Application Model développeur.

## Ressources connexes

- Pour plus d'informations sur la prise en AWS CodeBuild main, consultez [Commencer à AWS CodeBuild utiliser la console](#).
- Pour plus d'informations sur la résolution des problèmes dans CodeBuild, voir [Résolution des problèmes AWS CodeBuild](#).
- Pour plus d'informations sur les quotas dans CodeBuild, voir [Quotas pour AWS CodeBuild](#).

## Notices émanant de tiers AWS CodeBuild pour Windows

Lorsque vous utilisez des versions CodeBuild pour Windows, vous avez la possibilité d'utiliser des packages et modules tiers pour permettre à votre application créée de s'exécuter sur les systèmes d'exploitation Microsoft Windows et d'interagir avec certains produits tiers. La liste suivante contient les conditions d'utilisation légales tierces qui régissent votre utilisation des packages/modules tiers spécifiés.

### Rubriques

- [1\) image Docker de base — Windowsservercore](#)
- [2\) Image Docker basée sur Window—choco](#)
- [3\) Image Docker basée sur Windows : git --version 2.16.2](#)
- [4\) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools](#)
- [5\) Image Docker basée sur Windows : nuget.commandline --version 4.5.1](#)
- [7\) Image Docker basée sur Windows : netfx-4.6.2-devpack](#)
- [8\) Image Docker basée sur Windows — visualfsharpertools, version 4.0](#)
- [9\) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies](#)
- [10\) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1](#)
- [11\) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx](#)

- [12\) Image Docker basée sur Windows : dotnet-sdk](#)

## 1) image Docker de base — Windowsservercore

(les termes du contrat de licence sont disponibles à l'adresse suivante : [https://hub.docker.com/\\_/microsoft-windows-servercore](https://hub.docker.com/_/microsoft-windows-servercore))

Licence : Par la demande et l'utilisation de cette image de conteneur du système d'exploitation pour les conteneurs Windows, vous reconnaissez, comprenez et acceptez les conditions du contrat de licence complémentaire suivantes :

### CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

#### IMAGE DE CONTENEUR DU SYSTÈME D'EXPLOITATION

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) (référéncé en tant que « nous » ou « Microsoft ») vous accorde la licence complémentaire de cette image de conteneur du système d'exploitation (ci-après désigné « Complément »). Vous êtes autorisé à utiliser ce Complément conjointement avec le système d'exploitation hôte sous-jacent (ci-après désigné « Logiciel Hôte ») uniquement pour faciliter l'exécution des conteneurs du Logiciel Hôte. Les conditions du contrat de licence du Logiciel Hôte s'appliquent à votre utilisation du Complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le Logiciel Hôte. Vous pouvez utiliser ce Complément avec chaque copie sous licence valide du Logiciel Hôte.

#### EXIGENCES DE LICENCE SUPPLÉMENTAIRES DROITS AND/OR D'UTILISATION

Votre utilisation du Complément telle que spécifiée dans le paragraphe précédent peut entraîner la création ou la modification d'une image de conteneur (ci-après désigné « Image de conteneur ») qui inclut certains composants du Complément. Pour plus de clarté, une Image de conteneur est séparée et distincte d'une machine virtuelle ou d'une image d'appliance virtuelle. Conformément aux présentes conditions du contrat de licence, nous vous accordons un droit restreint de redistribuer lesdits composants du Complément sous les conditions suivantes :

- (i) vous pouvez utiliser les composants du Complément uniquement tels qu'ils sont utilisés dans, et dans le cadre de votre Image de conteneur,
- (ii) vous pouvez utiliser lesdits composants du Complément dans votre Image de conteneur tant que vous disposez de fonctionnalités principales importantes dans votre Image de conteneur qui sont matériellement séparées et distinctes du Complément ; et

(iii) vous acceptez d'inclure les présentes conditions du contrat de licence (ou des conditions semblables requises par nous ou un hôte) à votre Image de conteneur pour convenablement autoriser l'utilisation possible des composants du Complément par vos utilisateurs finaux.

Nous nous réservons tous les autres droits non accordés explicitement dans les présentes.

En utilisant le présent Complément, vous acceptez les présentes conditions. Si vous ne les acceptez pas, n'utilisez pas le présent Complément.

Dans le cadre des conditions de licence supplémentaires pour cette image de système d'exploitation de conteneur pour les conteneurs Windows, vous êtes également soumis aux termes de licence du logiciel hôte Windows Server sous-jacents, qui se trouvent à l'adresse : <https://www.microsoft.com/en-us/useterms>.

## 2) Image Docker basée sur Window—choco

(les termes du contrat de licence sont disponibles à l'adresse suivante : <https://github.com/chocolatey/choco/blob/master/LICENSE>)

Tous droits réservés 2011 - Present RealDimensions Software, LLC

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2.0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

## 3) Image Docker basée sur Windows : git --version 2.16.2

(les termes du contrat de licence sont disponibles à l'adresse suivante : <https://chocolatey.org/packages/git/2.16.2>)

Sous licence publique générale GNU, version 2, disponible à l'adresse : <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

## 4) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools

(les termes du contrat de licence sont disponibles à l'adresse : <https://www.visualstudio.com/license-terms/mt171552/>)

MICROSOFT VISUAL STUDIO 2015 EXTENSIONS, VISUAL STUDIO SHELLS et C++  
REDISTRIBUTABLE

-----

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Elles s'appliquent au logiciel désigné ci-dessus. Les conditions s'appliquent également à tous les services ou mises à jour Microsoft du logiciel, sauf dans la mesure où elles disposent de conditions supplémentaires.

-----

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS  
DISPOSEZ DES DROITS CI-DESSOUS.

1. INSTALLATION ET DROITS D'UTILISATION. Vous pouvez installer et utiliser n'importe quel nombre de copies du logiciel.
2. CONDITIONS POUR LES COMPOSANTS SPÉCIFIQUES.
  - a. Utilitaires. Le logiciel peut contenir certains éléments de la liste des utilitaires à l'adresse <https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs>. Vous pouvez copier et installer ces éléments, s'ils sont inclus dans le logiciel, sur les vôtres ou sur d'autres machines tierces, pour déboguer et déployer les applications et les bases de données que vous avez développées avec le logiciel. Veuillez noter que les Utilitaires sont conçus pour une utilisation temporaire, que Microsoft ne peut pas corriger ou mettre à jour les Utilitaires séparément du reste du logiciel, et que certains Utilitaires par nature peuvent permettre à d'autres d'accéder à des machines sur lesquelles ils sont installés. Par conséquent, vous devez supprimer tous les Utilitaires installés après la fin du débogage ou du déploiement de vos applications et bases de données. Microsoft n'est pas responsable de toute utilisation tierce ou de tout accès tiers à des Utilitaires installés sur n'importe quelle machine.
  - b. Plateformes Microsoft. Le logiciel peut inclure des composants provenant de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office et Microsoft SharePoint. Lesdits composants sont soumis à des accords séparés et leurs



propres politiques d'assistance produit, tel que décrit dans les conditions du contrat de licence disponibles dans le répertoire d'installation dudit composant ou dans le dossier « Licences » accompagnant le logiciel.

- c. Composants tiers. Le logiciel peut inclure des composants tiers soumis à des mentions légales distinctes ou régis par d'autres accords, comme décrit dans le ThirdPartyNotices fichier accompagnant le logiciel. Même si lesdits composants sont soumis à d'autres accords, les exclusions de responsabilités, les limitations et les exclusions concernant les dommages ci-dessous s'appliquent également. Le logiciel peut également inclure des composants sous licence de licences open source avec les obligations de disponibilité du code source. Des copies de ces licences, le cas échéant, sont incluses dans le ThirdPartyNotices fichier. Vous pouvez obtenir ledit code source grâce à nous, si nécessaire et requis par les licences open source applicables, en envoyant un mandat ou un chèque de 5 dollars américain à l'adresse : Source Code Compliance Team, Microsoft Corporation, 1 Microsoft Way, Redmond WA 98052, États-Unis. Veuillez écrire le code source d'un ou de plusieurs des composants répertoriés ci-dessous dans la ligne réservée aux notes de votre paiement :

- Remote Tools for Visual Studio 2015 ;
- Standalone Profiler for Visual Studio 2015 ;
- IntelliTraceCollector pour Visual Studio 2015 ;
- Microsoft VC++ Redistributable 2015 ;
- Multibyte MFC Library for Visual Studio 2015 ;
- Microsoft Build Tools 2015 ;
- Feedback Client ;
- Visual Studio 2015 Integrated Shell ; ou
- Visual Studio 2015 Isolated Shell.

Nous pouvons également effectuer une copie du code source disponible à l'adresse <http://thirdpartysource.microsoft.com>.

3. **DONNÉES.** Le logiciel peut récupérer des informations sur vous et votre utilisation du logiciel, et les envoyer à Microsoft. Microsoft peut utiliser ces informations pour fournir des services et améliorer nos produits et services. Vous pouvez vous désinscrire de la plupart desdits scénarios, mais pas de tous, tel que décrit dans la documentation du produit. Il existe également des fonctionnalités dans le logiciel qui peuvent vous permettre de récupérer des données d'utilisateurs de vos applications. Si vous utilisez ces fonctionnalités pour activer la récupération de données dans vos applications, vous devez respecter la loi en vigueur, notamment en avertissant de manière appropriée les utilisateurs de vos applications. Pour en savoir plus sur la collecte et

l'utilisation des données, consultez la documentation d'aide et la déclaration de confidentialité sur <https://privacy.microsoft.com/en-us/privacystatement>. Votre utilisation du logiciel sera considérée comme un consentement à ces pratiques.

4. **CHAMP D'APPLICATION DE LA LICENCE.** Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Vous ne pouvez pas
  - contourner n'importe quelles limitations du logiciel ;
  - procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, ou essayer de le faire, sauf si exigé et uniquement dans la mesure requise par les conditions du contrat de licence tierces régissant l'utilisation de certains composants open source pouvant être inclus avec le logiciel ;
  - supprimer, minimiser, bloquer ou modifier toutes les mentions de Microsoft ou de ses fournisseurs dans le logiciel ;
  - utiliser le logiciel de manière illégale ; ou
  - partager, publier, louer ou céder en location le logiciel, ou proposer le logiciel en tant que solution hébergée autonome pour que d'autres personnes l'utilisent.
5. **RESTRICTIONS À L'EXPORTATION.** Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel, qui incluent des restrictions sur des destinations, des utilisateurs finaux, et des utilisations finales. Pour plus d'informations sur les restrictions à l'exportation, visitez ([aka.ms/exporting](https://aka.ms/exporting)).
6. **SERVICES D'ASSISTANCE.** Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
7. **ACCORD ENTIER.** Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et les services d'assistance.
8. **LOI APPLICABLE.** Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington s'appliquent à l'interprétation des réclamations et aux réclamations pour violation du présent accord, et les lois de l'État où vous résidez s'appliquent à toutes les autres réclamations. Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois du pays s'appliquent.
9. **DROITS DES CONSOMMATEURS ; DIFFÉRENCES RÉGIONALES.** Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits, notamment les droits des consommateurs,

conformément aux lois de votre État ou pays. À part et en dehors de votre relation avec Microsoft, vous pouvez également disposer de droits concernant la partie grâce à laquelle vous avez acquis le logiciel. Cet accord ne modifie pas lesdits autres droits si les lois de votre État ou pays ne permettent pas à le faire. Par exemple, si vous avez acquis le logiciel dans l'une des régions ci-dessous, ou que la loi obligatoire du pays s'applique, les dispositions suivantes s'appliquent à vous :

- a. Australie. Vous disposez de garanties statutaires conformément à l'Australian Consumer Law et rien dans le présent accord n'est destiné à affecter lesdits droits.
- b. Canada. Si vous avez acquis le présent logiciel au Canada, vous pouvez cesser de recevoir des mises à jour en désactivant la fonctionnalité de mise à jour automatique, déconnecter votre appareil d'Internet (toutefois, si et quand vous le reconnectez à Internet, le logiciel reprend la vérification et l'installation des mises à jour) ou désinstaller le logiciel. La documentation produit, le cas échéant, peut également spécifier la façon de désactiver les mises à jour d'un appareil ou logiciel spécifique.
- c. Allemagne et Autriche.
  - i. Garantie. Le logiciel sous licence valide logiciel fonctionnera tel que décrit dans n'importe quel matériel Microsoft accompagnant le logiciel. Toutefois, Microsoft n'offre aucune garantie contractuelle concernant le logiciel sous licence.
  - ii. Limitation de responsabilité. En cas de comportements intentionnels, négligence grave, réclamations reposant sur le Product Liability Act, ainsi qu'en cas de mort ou de blessures personnelles ou physiques, Microsoft sera tenu responsable conformément à la loi statutaire. Sous condition de la clause précédente (ii), Microsoft sera uniquement tenu responsable en cas de négligence mineure si Microsoft ne respecte pas lesdites obligations contractuelles matérielles, dont l'accomplissement faciliterait l'exécution parfaite du présent accord, dont le non-respect pourrait menacer l'objet même du présent accord et dont la conformité avec une partie peut constamment renforcer la confiance (ci-après désigné « Obligations cardinales »). Dans d'autres cas de négligence mineure, Microsoft ne sera pas tenu responsable.

10 EXCLUSION DE GARANTIE. LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

11 LIMITATION ET EXCLUSION DES DOMMAGES. MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES

DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES. La présente limitation s'applique (a) à tout aspect lié au logiciel, aux services (notamment le code) sur les sites Internet tiers, ou les applications tierces ; et (b) aux revendications pour violation du contrat, violation de garantie, garantie ou condition, responsabilité stricte, négligence, ou tout autre délit dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

ID EULA : VS2015 \_Update3\_ \_ ShellsRedist <ENU>

## 5) Image Docker basée sur Windows : nuget.commandline --version 4.5.1

(les termes du contrat de licence sont disponibles à l'adresse : <https://github.com/NuGet/Home/blob/dev/LICENSE.txt>)

Copyright (c) .NET Foundation. Tous droits réservés.

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2,0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

## 7) Image Docker basée sur Windows : netfx-4.6.2-devpack

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

.NET FRAMEWORK ET LES PACKAGES DE LANGUE ASSOCIÉS POUR LE SYSTÈME D'EXPLOITATION MICROSOFT WINDOWS

-----

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) vous accorde la licence du présent complément. Si vous êtes sous licence pour utiliser le logiciel du système d'exploitation Microsoft Windows (ci-après désigné le « logiciel »), vous pouvez utiliser le présent complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le logiciel. Vous pouvez utiliser le présent Complément avec chaque copie sous licence valide du logiciel.

Les conditions du contrat de licence suivantes décrivent les conditions d'utilisation supplémentaires du présent complément. Les présentes conditions et conditions du contrat de licence du logiciel s'appliquent à votre utilisation du complément. En cas de conflit, les présentes conditions du contrat de licence complémentaire s'appliquent.

EN UTILISANT CE SUPPLÉMENT, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES ACCEPTEZ PAS, N'UTILISEZ PAS CE SUPPLÉMENT.

-----

Si vous respectez les présentes conditions du contrat de licence, vous disposez des droits ci-dessous.

1. CODE DISTRIBUABLE. Le supplément est composé d'un code distribuable. Un « Code Distribuable » est un code que vous êtes autorisé à distribuer dans les programmes que vous développez si vous respectez les conditions ci-dessous.
  - a. Droit d'utilisation et de distribution.
    - Vous pouvez copier et distribuer le complément sous forme de code objet.
    - Distribution tierce. Vous pouvez autoriser les distributeurs de vos programmes à copier et à distribuer le Code Distribuable dans le cadre de ces programmes.
  - b. Exigences relatives à la distribution. Pour tout code distribuable que vous distribuez, vous devez
    - lui ajouter des fonctionnalités principales importantes dans vos programmes ;
    - pour n'importe quel Code Distribuable disposant d'une extension de fichier .lib, distribuer uniquement les résultats de l'exécution dudit Code Distribuable via un éditeur de liens avec votre programme ;
    - distribuer le Code Distribuable inclus dans un programme d'installation uniquement dans le cadre de ce programme d'installation sans modification ;
    - exiger que les distributeurs et les utilisateurs finaux externes acceptent les conditions qui le protègent au moins autant que le présent accord.
    - afficher vos mentions des droits d'auteur valides sur vos programmes ; et

- indemniser, défendre et exonérer Microsoft face à toutes les revendications, y compris les frais d'avocat, liés à la distribution ou à l'utilisation de vos programmes.

c. Restrictions de distribution. Il se peut que vous ne

- modifier les droits d'auteur, les marques déposées ou les brevets du Code Distribuable ;
- utiliser des marques déposées de Microsoft dans vos noms de programme ou en laissant croire que vos programmes proviennent de Microsoft ou sont approuvés par Microsoft ;
- distribuer un Code Distribuable à exécuter sur une plateforme autre que Windows ;
- inclure un Code Distribuable dans des programmes malveillants, frauduleux ou illicites ; ou
- modifier ou distribuer le code source de n'importe quel Code Distribuable afin que toute partie de celui-ci soit soumise à une licence d'exclusion. Une licence d'exclusion exige, comme condition d'utilisation, modification ou distribution, que
  - le code divulgué ou distribué sous forme de code source ; ou
  - d'autres aient le droit de la modifier.

2. SERVICES D'ASSISTANCE POUR LE COMPLÉMENT. Microsoft fournit des services de support pour ce logiciel, comme décrit sur le [site www.support.microsoft.com/common/international.aspx](http://www.support.microsoft.com/common/international.aspx).

## 8) Image Docker basée sur Windows — visualfsharpertools, version 4.0

(les termes du contrat de licence sont disponibles à l'adresse : <https://github.com/dotnet/fsharp/blob/main/License.txt>)

Copyright (c) Microsoft Corporation. Tous droits réservés.

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2,0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

## 9) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies

### CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE MICROSOFT

## ASSEMBLAGES DE RÉFÉRENCE DE LA BIBLIOTHÈQUE DE CLASSE PORTABLE MICROSOFT .NET – 4.6

-----

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Veuillez les lire. Elles s'appliquent au logiciel désigné ci-dessus. Les conditions s'appliquent également à l'ensemble des

- mises à jour,
- compléments,
- services Internet, et
- services d'assistance Microsoft

pour ce logiciel, sauf si d'autres conditions accompagnent lesdits éléments. Si tel est le cas, lesdites conditions s'appliquent.

EN UTILISANT LE LOGICIEL, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES  
ACCEPTEZ PAS, N'UTILISEZ PAS LE LOGICIEL.

-----

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS  
DISPOSEZ DES DROITS PERPÉTUELS CI-DESSOUS.

1. INSTALLATION ET DROITS D'UTILISATION. Vous pouvez installer et utiliser n'importe quel nombre de copies du logiciel pour concevoir, développer et tester vos programmes.
2. EXIGENCES DE LICENCE SUPPLÉMENTAIRES : DROITS AND/OR D'UTILISATION.
  - a. Code Distribuible. Vous pouvez distribuer le logiciel dans les programmes d'outils de développement que vous développez, pour permettre aux clients de vos programmes de développer des bibliothèques portables à utiliser avec n'importe quel appareil ou système d'exploitation, si vous respectez les conditions ci-dessous.
    - i. Droit d'utilisation et de distribution. Le logiciel s'appelle « Code distribuible ».
      - Code Distribuible. Vous pouvez copier et distribuer le logiciel sous forme de code objet.
      - Distribution tierce. Vous pouvez autoriser les distributeurs de vos programmes à copier et à distribuer le Code Distribuible dans le cadre de ces programmes.

ii. Exigences relatives à la distribution. Pour tout code distribuable que vous distribuez, vous devez

- lui ajouter des fonctionnalités principales importantes dans vos programmes ;
- exiger que les distributeurs et vos clients acceptent les conditions qui le protègent au moins autant que le présent accord.
- afficher vos mentions des droits d'auteur valides sur vos programmes ; et
- indemniser, défendre et exonérer Microsoft face à toutes les revendications, y compris les frais d'avocat, liés à la distribution ou à l'utilisation de vos programmes.

iii. Restrictions de distribution. Il se peut que vous ne

- modifier les droits d'auteur, les marques déposées ou les brevets du Code Distribuable ;
- utiliser des marques déposées de Microsoft dans vos noms de programme ou en laissant croire que vos programmes proviennent de Microsoft ou sont approuvés par Microsoft ;
- inclure un Code Distribuable dans des programmes malveillants, frauduleux ou illicites ; ou
- modifier ou distribuer le Code Distribuable afin que toute partie de celui-ci soit soumise à une licence d'exclusion. Une licence d'exclusion exige, comme condition d'utilisation, modification ou distribution, que
  - le code divulgué ou distribué sous forme de code source ; ou
  - d'autres aient le droit de la modifier.

3. CHAMP D'APPLICATION DE LA LICENCE. Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Vous ne pouvez pas

- contourner n'importe quelles limitations du logiciel ;
- procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, sauf si exigé et uniquement dans la mesure autorisée expressément par la législation applicable, malgré la présente limitation ;
- publier le logiciel pour que d'autres le copient ; ou
- louer, céder en location ou prêter le logiciel.

4. COMMENTAIRES. Vous pouvez fournir des commentaires sur le logiciel. Si vous communiquer des commentaires sur le logiciel à Microsoft, vous accordez à Microsoft, sans frais, le droit d'utiliser, de partager et de commercialiser vos commentaires de quelque façon que ce soit.



et à quelque fin que ce soit. Vous accordez également à des tiers, sans frais, les droits de brevet nécessaires à leurs produits, technologies et services afin d'utiliser ou de communiquer avec n'importe quelle partie spécifique d'un logiciel ou d'un service Microsoft qui inclut les commentaires. Vous ne pourrez pas communiquer de commentaires soumis à une licence qui exige que Microsoft concède des licences pour son logiciel ou sa documentation à un tiers, car vos commentaires y sont inclus. Ces droits survivent au présent accord.

5. TRANSFERT À UN TIERS. Le premier utilisateur du logiciel peut le transférer, ainsi que le présent accord, directement à un tiers. Avant le transfert, ce tiers doit accepter que le présent accord s'applique au transfert et à l'utilisation du logiciel. Le premier utilisateur doit désinstaller le logiciel avant de le transférer séparément depuis l'appareil. Le premier utilisateur ne peut conserver aucune copie.
6. RESTRICTIONS À L'EXPORTATION. Le logiciel est soumis lois et réglementations d'exportation américaines. Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel. Lesdites lois incluent des restrictions sur des destinations, des utilisateurs finaux, et des utilisations finales. Pour plus d'informations, consultez [www.microsoft.com/exporting](http://www.microsoft.com/exporting).
7. SERVICES D'ASSISTANCE. Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
8. ACCORD ENTIER. Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et tous les services d'assistance que nous fournissons.
9. LOI APPLICABLE.
  - a. États-Unis. Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington régissent l'interprétation du présent accord et s'appliquent aux réclamations pour violation de ce dernier, indépendamment de toute règle de conflit de lois. Les lois de l'État dans lequel vous résidez régissent toutes les autres réclamations, notamment les réclamations en vertu des lois sur la protection des consommateurs, des lois sur la concurrence déloyale, et des lois sur la responsabilité civile.
  - b. En dehors des États-Unis. Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois de ce dernier s'appliquent.
10. CONSÉQUENCE LÉGALE. Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits en vertu des lois de votre pays. Vous pouvez également disposer de droits liés au tiers grâce auquel vous avez acquis le logiciel. Le présent accord ne modifie pas vos droits en vertu des lois de votre pays si les lois de votre pays ne permettent pas à le faire.

11 EXCLUSION DE GARANTIE. LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. VOUS POUVEZ DISPOSER DE DROITS DES CONSOMMATEURS SUPPLÉMENTAIRES OU DE GARANTIES STATUTAIRES EN VERTU DE VOS LOIS LOCALES QUE LE PRÉSENT ACCORD NE PEUT PAS MODIFIER. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

POUR L'AUSTRALIE—VOUS DISPOSEZ DE GARANTIES STATUTAIRES CONFORMÉMENT À L'AUSTRALIAN CONSUMER LAW ET RIEN DANS LE PRÉSENT ACCORD N'EST DESTINÉ À AFFECTER LESDITS DROITS.

12 LIMITATION ET EXCLUSION DES RECOURS ET DES DOMMAGES-INTÉRÊTS. MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES.

Cette limitation s'applique

- à tout ce qui concerne le logiciel, les services, le contenu (notamment le code) sur les sites Internet tiers, ou les programmes tiers ; et
- aux réclamations pour violation de contrat, violation de garantie ou de condition, responsabilité stricte, négligence, ou autres délits dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

## 10) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1

(les termes du contrat de licence sont disponibles à l'adresse : <https://www.visualstudio.com/license-terms/mt644918/>)

OUTILS DE GÉNÉRATION MICROSOFT VISUAL C++

## CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE MICROSOFT

### OUTILS DE GÉNÉRATION MICROSOFT VISUAL C++

-----

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Elles s'appliquent au logiciel désigné ci-dessus. Les conditions s'appliquent également à tous les services ou mises à jour Microsoft du logiciel, sauf dans la mesure où elles disposent de conditions différentes.

-----

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS DISPOSEZ DES DROITS CI-DESSOUS.

#### 1. INSTALLATION ET DROITS D'UTILISATION.

a. Un utilisateur peut utiliser des copies du logiciel pour développer et tester ses applications.

2. DONNÉES. Le logiciel peut récupérer des informations sur vous et votre utilisation du logiciel, et les envoyer à Microsoft. Microsoft peut utiliser ces informations pour fournir des services et améliorer nos produits et services. Vous pouvez vous désinscrire de la plupart desdits scénarios, mais pas de tous, tel que décrit dans la documentation du produit. Il existe également des fonctionnalités dans le logiciel qui peuvent vous permettre de récupérer des données d'utilisateurs de vos applications. Si vous utilisez ces fonctionnalités pour activer la récupération de données dans vos applications, vous devez respecter la loi en vigueur, notamment en avertissant de manière appropriée les utilisateurs de vos applications. Vous pouvez en savoir plus sur la récupération et l'utilisation des données dans la documentation d'aide et la déclaration de confidentialité à l'adresse <http://go.microsoft.com/fwlink/?LinkID=528096>. Votre utilisation du logiciel sera considérée comme un consentement à ces pratiques.

#### 3. CONDITIONS POUR LES COMPOSANTS SPÉCIFIQUES.

a. Build Server. Le logiciel peut contenir certains composants du serveur de compilation répertoriés dans des fichiers BuildServer .TXT, and/or tous les fichiers répertoriés dans la BuildServer liste étant situés conformément aux présentes conditions de licence logicielle Microsoft. Vous pouvez copier et installer ces éléments, s'ils sont inclus dans le logiciel, sur vos machines de génération. Vous et d'autres utilisateurs de votre organisation peuvent utiliser ces éléments sur vos machines de génération uniquement à des fins de compilation, de génération, de vérification et d'archivage de vos applications ou d'exécution de tests de qualité ou de performance dans le cadre du processus de génération.

- b. Plateformes Microsoft. Le logiciel peut inclure des composants provenant de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office et Microsoft SharePoint. Lesdits composants sont soumis à des accords séparés et leurs propres politiques d'assistance produit, tel que décrit dans les conditions du contrat de licence disponibles dans le répertoire d'installation dudit composant ou dans le dossier « Licences » accompagnant le logiciel.
  - c. Composants tiers. Le logiciel peut inclure des composants tiers soumis à des mentions légales distinctes ou régis par d'autres accords, comme décrit dans le ThirdPartyNotices fichier accompagnant le logiciel. Même si lesdits composants sont soumis à d'autres accords, les exclusions de responsabilités, les limitations et les exclusions concernant les dommages ci-dessous s'appliquent également.
  - d. Gestionnaires de package. Le logiciel peut inclure des gestionnaires de package, tels que Nuget, qui vous permettent de télécharger d'autres packages logiciels Microsoft et des packages logiciels tiers à utiliser avec votre application. Ces packages sont soumis à leurs propres licences, et non pas au présent accord. Microsoft ne permet pas de distribuer, d'autoriser ou de fournir des garanties pour l'un des packages tiers.
4. CHAMP D'APPLICATION DE LA LICENCE. Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Pour plus d'informations, consultez [#1](https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms) - installation-and-use-rights Vous ne pouvez pas
- contourner n'importe quelles limitations du logiciel ;
  - procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, ou essayez de le faire, sauf si exigé et uniquement dans la mesure requise par les conditions du contrat de licence tierces régissant l'utilisation de certains composants open source pouvant être inclus avec le logiciel ;
  - supprimer, minimiser, bloquer ou modifier toutes les mentions de Microsoft ou de ses fournisseurs ;
  - utiliser le logiciel de manière illégale ; ou
  - partager, publier, louer ou céder en location le logiciel, ou proposer le logiciel en tant que solution hébergée autonome pour que d'autres personnes l'utilisent.

5. **RESTRICTIONS À L'EXPORTATION.** Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel, qui incluent des restrictions sur des destinations, des utilisateurs finaux et des utilisations finales. Pour plus d'informations sur les restrictions à l'exportation, visitez ([aka.ms/exporting](https://aka.ms/exporting)).
6. **SERVICES D'ASSISTANCE.** Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
7. **ACCORD ENTIER.** Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et les services d'assistance.
8. **LOI APPLICABLE.** Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington s'appliquent à l'interprétation des réclamations et aux réclamations pour violation du présent accord, et les lois de l'État où vous résidez s'appliquent à toutes les autres réclamations. Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois du pays s'appliquent.
9. **DROITS DES CONSOMMATEURS ; DIFFÉRENCES RÉGIONALES.** Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits, notamment les droits des consommateurs, conformément aux lois de votre État ou pays. À part et en dehors de votre relation avec Microsoft, vous pouvez également disposer de droits concernant la partie grâce à laquelle vous avez acquis le logiciel. Cet accord ne modifie pas lesdits autres droits si les lois de votre État ou pays ne permettent pas à le faire. Par exemple, si vous avez acquis le logiciel dans l'une des régions ci-dessous, ou que la loi obligatoire du pays s'applique, les dispositions suivantes s'appliquent à vous :
  - **Australie.** Vous disposez de garanties statutaires conformément à l'Australian Consumer Law et rien dans le présent accord n'est destiné à affecter lesdits droits.
  - **Canada.** Si vous avez acquis le présent logiciel au Canada, vous pouvez cesser de recevoir des mises à jour en désactivant la fonctionnalité de mise à jour automatique, déconnecter votre appareil d'Internet (toutefois, si et quand vous le reconnectez à Internet, le logiciel reprend la vérification et l'installation des mises à jour) ou désinstaller le logiciel. La documentation produit, le cas échéant, peut également spécifier la façon de désactiver les mises à jour d'un appareil ou logiciel spécifique.
  - **Allemagne et Autriche.**
    - **Garantie.** Le logiciel sous licence valide logiciel fonctionnera tel que décrit dans n'importe quel matériel Microsoft accompagnant le logiciel. Toutefois, Microsoft n'offre aucune garantie contractuelle concernant le logiciel sous licence.

- Limitation de responsabilité. En cas de comportements intentionnels, négligence grave, réclamations reposant sur le Product Liability Act, ainsi qu'en cas de mort ou de blessures personnelles ou physiques, Microsoft sera tenu responsable conformément à la loi statutaire.

Sous condition de la clause précédente (ii), Microsoft sera uniquement tenu responsable en cas de négligence mineure si Microsoft ne respecte pas lesdites obligations contractuelles matérielles, dont l'accomplissement faciliterait l'exécution parfaite du présent accord, dont le non-respect pourrait menacer l'objet même du présent accord et dont la conformité avec une partie peut constamment renforcer la confiance (ci-après désigné « Obligations cardinales »).

Dans d'autres cas de négligence mineure, Microsoft ne sera pas tenu responsable.

**10. CONSÉQUENCE LÉGALE.** Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits en vertu des lois de votre État ou pays. Le présent accord ne modifie pas vos droits en vertu des lois de votre État ou pays si les lois de votre État ou pays ne permettent pas à le faire. Sans restreindre la portée générale de ce qui précède, pour l'Australie, VOUS DISEPOSEZ DE GARANTIES STATUTAIRES CONFORMÉMENT À L'AUSTRALIAN CONSUMER LAW ET RIEN DANS LE PRÉSENT ACCORD N'EST DESTINÉ À AFFECTER LESDITS DROITS

**11. EXCLUSION DE GARANTIE.** LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

**12. LIMITATION ET EXCLUSION DES DOMMAGES.** MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES.

La présente limitation s'applique (a) à tout aspect lié au logiciel, aux services (notamment le code) sur les sites Internet tiers, ou les applications tierces ; et (b) aux revendications pour violation du contrat, violation de garantie, garantie ou condition, responsabilité stricte, négligence, ou tout autre délit dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

## 11) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx

### CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

#### MICROSOFT .NET FRAMEWORK 3.5 SP1 POUR SYSTÈME D'EXPLOITATION MICROSOFT WINDOWS

-----

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) vous accorde la licence du présent complément. Si vous êtes sous licence pour utiliser le logiciel du système d'exploitation Microsoft Windows (pour lequel le présent complément s'applique) (ci-après désigné le « logiciel »), vous pouvez utiliser le présent complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le logiciel. Vous pouvez utiliser une copie du présent Complément avec chaque copie sous licence valide du logiciel.

Les conditions du contrat de licence suivantes décrivent les conditions d'utilisation supplémentaires du présent complément. Les présentes conditions et conditions du contrat de licence du logiciel s'appliquent à votre utilisation du complément. En cas de conflit, les présentes conditions du contrat de licence complémentaire s'appliquent.

EN UTILISANT CE SUPPLÉMENT, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES ACCEPTEZ PAS, N'UTILISEZ PAS CE SUPPLÉMENT.

-----

Si vous respectez les présentes conditions du contrat de licence, vous disposez des droits ci-dessous.

1. SERVICES D'ASSISTANCE POUR LE COMPLÉMENT. Microsoft fournit des services de support pour ce logiciel, comme décrit sur le [site www.support.microsoft.com/common/international.aspx](http://www.support.microsoft.com/common/international.aspx).
2. TESTS D'ÉVALUATION MICROSOFT .NET. Le logiciel inclut .NET Framework, Windows Communication Foundation, Windows Presentation Foundation, et les composants Windows Workflow Foundation des systèmes d'exploitation Windows (.NET Components). Vous pouvez effectuer des tests d'évaluation internes de .NET Components. Vous pouvez communiquer les résultats de n'importe quel test d'évaluation des .NET Components, à condition que vous respectiez les conditions stipulées à l'adresse <http://go.microsoft.com/fwlink/?LinkID=66406>.



Nonobstant tout autre accord avec Microsoft, si vous communiquez les résultats des tests d'évaluation, Microsoft se réserve le droit de communiquer les résultats des tests d'évaluation qu'il réalise pour vos produits qui concurrencent le .NET Component applicable, pourvu qu'il respecte les mêmes conditions stipulées à l'adresse <http://go.microsoft.com/fwlink/?LinkID=66406>.

## 12) Image Docker basée sur Windows : dotnet-sdk

(disponible au <https://github.com/dotnet/core/blob/main/LICENSEformat.TXT>)

La licence MIT (MIT)

Copyright (c) Microsoft Corporation

L'autorisation est accordée, gratuitement, à toute personne obtenant une copie de ce logiciel et des fichiers de documentation associés (le « Logiciel »), d'utiliser le Logiciel sans restriction, y compris, sans s'y limiter, le droit d'utiliser, de copier, de modifier, de fusionner, de publier, de distribuer, de sous-licencier, de and/or vendre des copies du Logiciel, et d'autoriser les personnes auxquelles le Logiciel est fourni à le faire, sous réserve des conditions suivantes :

Les mentions des droits d'auteur et de ladite permission susmentionnées doivent être incluses dans toutes les copies ou parties importantes du Logiciel.

LE LOGICIEL EST FOURNI « EN L'ÉTAT », SANS GARANTIE AUCUNE, EXPRESSE OU IMPLICITE, Y COMPRIS SANS S'Y LIMITER LA GARANTIE DE QUALITE MARCHANDE, D'ADEQUATION A UN OBJECTIF PARTICULIER ET DE NON-INFRACTION. Tous droits réservés.

## Utiliser des clés de CodeBuild condition comme variables de rôle de service IAM pour contrôler l'accès au build

Avec l'ARN de CodeBuild construction, vous pouvez restreindre l'accès aux ressources de construction en utilisant des clés contextuelles pour limiter l'accès aux ressources dans votre rôle CodeBuild de service. En CodeBuild effet, les clés qui peuvent être utilisées pour contrôler le comportement d'accès aux builds sont `codebuild:buildArn` et `codebuild:projectArn`. Avec l'ARN du projet de construction, vous pouvez vérifier si un appel à votre ressource provient d'un projet de construction spécifique. Pour vérifier cela, utilisez les clés de `codebuild:projectArn` condition `codebuild:buildArn` ou dans une politique basée sur l'identité IAM.



Pour utiliser les clés de `codebuild:projectArn` condition `codebuild:buildArn` ou dans votre politique, incluez-les comme condition avec l'un des opérateurs de condition ARN. La valeur de la clé doit être une variable IAM qui se résout en un ARN valide. Dans l'exemple de politique ci-dessous, le seul accès autorisé sera le projet de construction avec l'ARN du projet pour la variable `${codebuild:projectArn}` IAM.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor0",
 "Effect": "Allow",
 "Action": "s3:PutObject",
 "Resource": "arn:aws:s3:::bucket-name/${codebuild:projectArn}/*"
 }
]
}
```

## AWS CodeBuild clés de condition

AWS CodeBuild fournit un ensemble de clés de condition que vous pouvez utiliser dans les politiques IAM pour appliquer les politiques de votre organisation sur les CodeBuild ressources telles que les projets et les flottes. Les clés de condition couvrent la plupart des contextes de demande d'API CodeBuild, notamment les paramètres réseau, les configurations d'informations d'identification et les restrictions de calcul.

### Rubriques

- [Appliquez les paramètres de connectivité VPC à vos projets et à vos flottes](#)
- [Empêchez les modifications non autorisées des spécifications de construction du projet](#)
- [Limitez les types de calcul pour vos builds](#)
- [Réglages variables de l'environnement de contrôle](#)
- [Utiliser des variables dans les noms des clés de condition](#)
- [Vérifier l'existence d'attributs dans les demandes d'API](#)

## Appliquez les paramètres de connectivité VPC à vos projets et à vos flottes

Cette politique permet à l'appelant d'utiliser les sous-réseaux et les groupes de sécurité sélectionnés VPCs lors de la création de CodeBuild projets et de flottes. Pour plus d'informations sur les clés de contexte à valeurs multiples, voir Clés de contexte [à valeur unique ou clés de contexte à valeurs multiples](#).

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Action": [
 "codebuild:CreateProject",
 "codebuild:CreateFleet"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "codebuild:vpcConfig.vpcId": [
 "vpc-01234567890abcdef",
 "vpc-abcdef01234567890"
],
 "codebuild:vpcConfig.subnets": [
 "subnet-1234abcd",
 "subnet-5678abcd"
],
 "codebuild:vpcConfig.securityGroupIds": [
 "sg-12345678abcdefghij",
 "sg-01234567abcdefghij"
]
 }
 }
]
}
```

## Empêchez les modifications non autorisées des spécifications de construction du projet

Cette politique ne permet pas à l'appelant de remplacer les spécifications de construction dans le champ `buildspecOverride`

### Note

La clé de `codebuild:source.buildspec` condition prend uniquement en charge l'opérateur Null pour vérifier l'existence du champ API. Il n'évalue pas le contenu de la spécification de construction.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Action": "codebuild:StartBuild",
 "Resource": "*"
 }, {
 "Effect": "Deny",
 "Action": "codebuild:StartBuild",
 "Resource": "*",
 "Condition": {
 "Null": {
 "codebuild:source.buildspec": "false"
 }
 }
 }]
}
```

## Limitez les types de calcul pour vos builds

Cette politique permet de créer des flottes capables de construire uniquement avec `c5.large` ou de `m5.large` [calculer le type d'instance](#).

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Action": "codebuild:CreateFleet",
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "codebuild:computeConfiguration.instanceType": ["c5.large",
"m5.large"]
 }
 }
 }]
}
```

## Réglages variables de l'environnement de contrôle

Cette politique permet à l'appelant de remplacer la variable d'STAGEEnvironnement par l'une ou l'autreBETA. GAMMA II nie également explicitement le remplacement par STAGE être et nie PRODUCTION le remplacement de la variable d'MY\_APP\_VERSIONenvironnement. Pour les clés de contexte à valeurs multiples, consultez la section Clés de [contexte à valeur unique ou à valeurs multiples](#).

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codebuild:StartBuild"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "codebuild:environment.environmentVariables/STAGE.value": [
```

```

 "BETA",
 "GAMMA"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:StartBuild"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "codebuild:environment.environmentVariables/STAGE.value":
"PRODUCTION"
 },
 "ForAnyValue:StringEquals": {
 "codebuild:environment.environmentVariables.name": [
 "MY_APP_VERSION"
]
 }
 }
 }
]
}

```

## Utiliser des variables dans les noms des clés de condition

Vous pouvez utiliser des variables dans les noms de clés de condition, tels que `secondarySources/${sourceIdentifiant}.location` et `secondaryArtifacts/${artifactIdentifiant}.location`, où vous pouvez spécifier votre [source](#) secondaire ou votre identifiant d'[artefact](#) secondaire dans la politique IAM. La politique ci-dessous permet à l'appelant de créer un projet avec un emplacement source spécifique pour la source `mySecondSource` secondaire.

JSON

```

{
 "Version": "2012-10-17",

```

```

 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateProject",
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "codebuild:secondarySources/mySecondSource.location": "my-
source-location"
 }
 }
 }
]
 }
}

```

## Vérifier l'existence d'attributs dans les demandes d'API

CodeBuild prend en charge les clés de condition pour vérifier l'existence de certains champs dans la demande d'API. La politique applique l'exigence VPC lors de la création ou de la mise à jour de projets.

```

{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Action": [
 "codebuild:CreateProject",
 "codebuild:UpdateProject"
],
 "Resource": "*",
 "Condition": {
 "Null": {
 "codebuild:vpcConfig.vpcId": "false"
 }
 }
 }]
}

```

# Exemples de code à CodeBuild utiliser AWS SDKs

Les exemples de code suivants montrent comment utiliser CodeBuild un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

## Exemples de code

- [Exemples de base pour CodeBuild l'utilisation AWS SDKs](#)
  - [Actions d' CodeBuild utilisation AWS SDKs](#)
    - [Utilisation CreateProject avec un AWS SDK ou une CLI](#)
    - [Utilisation ListBuilds avec un AWS SDK ou une CLI](#)
    - [Utilisation ListProjects avec un AWS SDK ou une CLI](#)
    - [Utilisation StartBuild avec un AWS SDK ou une CLI](#)

# Exemples de base pour CodeBuild l'utilisation AWS SDKs

Les exemples de code suivants montrent comment utiliser les principes de base de AWS CodeBuild with AWS SDKs.

## Exemples

- [Actions d' CodeBuild utilisation AWS SDKs](#)
  - [Utilisation CreateProject avec un AWS SDK ou une CLI](#)
  - [Utilisation ListBuilds avec un AWS SDK ou une CLI](#)
  - [Utilisation ListProjects avec un AWS SDK ou une CLI](#)
  - [Utilisation StartBuild avec un AWS SDK ou une CLI](#)

## Actions d' CodeBuild utilisation AWS SDKs

Les exemples de code suivants montrent comment effectuer des CodeBuild actions individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez la [Référence des API AWS CodeBuild](#).

### Exemples

- [Utilisation CreateProject avec un AWS SDK ou une CLI](#)
- [Utilisation ListBuilds avec un AWS SDK ou une CLI](#)
- [Utilisation ListProjects avec un AWS SDK ou une CLI](#)
- [Utilisation StartBuild avec un AWS SDK ou une CLI](#)

### Utilisation **CreateProject** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser CreateProject.

#### CLI

##### AWS CLI

Exemple 1 : pour créer un projet AWS CodeBuild de construction

L'create-projectexemple suivant crée un projet de CodeBuild construction à l'aide des fichiers source d'un compartiment S3.

```
aws codebuild create-project \
 --name "my-demo-project" \
 --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source.zip\"}" \
 --artifacts "{\"type\": \"S3\", \"location\": \"codebuild-us-
west-2-123456789012-output-bucket\"}" \
 --environment "{\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
 --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```



## Sortie :

```
{
 "project": {
 "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
 "name": "my-cli-demo-project",
 "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
 "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
 "lastModified": 1556839783.274,
 "badge": {
 "badgeEnabled": false
 },
 "queuedTimeoutInMinutes": 480,
 "environment": {
 "image": "aws/codebuild/standard:1.0",
 "computeType": "BUILD_GENERAL1_SMALL",
 "type": "LINUX_CONTAINER",
 "imagePullCredentialsType": "CODEBUILD",
 "privilegedMode": false,
 "environmentVariables": []
 },
 "artifacts": {
 "location": "codebuild-us-west-2-123456789012-output-bucket",
 "name": "my-cli-demo-project",
 "namespaceType": "NONE",
 "type": "S3",
 "packaging": "NONE",
 "encryptionDisabled": false
 },
 "source": {
 "type": "S3",
 "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip",
 "insecureSsl": false
 },
 "timeoutInMinutes": 60,
 "cache": {
 "type": "NO_CACHE"
 },
 "created": 1556839783.274
 }
}
```

```
}
```

Exemple 2 : pour créer un projet de AWS CodeBuild construction à l'aide d'un fichier d'entrée JSON pour les paramètres

L'`create-project` exemple suivant crée un projet de CodeBuild construction en transmettant tous les paramètres requis dans un fichier d'entrée JSON. Créez le modèle de fichier d'entrée en exécutant la commande avec uniquement le `--generate-cli-skeleton` parameter.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Le fichier d'entrée JSON `create-project.json` contient le contenu suivant :

```
{
 "name": "codebuild-demo-project",
 "source": {
 "type": "S3",
 "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
 },
 "artifacts": {
 "type": "S3",
 "location": "codebuild-region-ID-account-ID-output-bucket"
 },
 "environment": {
 "type": "LINUX_CONTAINER",
 "image": "aws/codebuild/standard:1.0",
 "computeType": "BUILD_GENERAL1_SMALL"
 },
 "serviceRole": "serviceIAMRole"
}
```

Sortie :

```
{
 "project": {
 "name": "codebuild-demo-project",
 "serviceRole": "serviceIAMRole",
 "tags": [],
 "artifacts": {
 "packaging": "NONE",
 "type": "S3",
 "location": "codebuild-region-ID-account-ID-output-bucket",

```

```

 "name": "message-util.zip"
 },
 "lastModified": 1472661575.244,
 "timeoutInMinutes": 60,
 "created": 1472661575.244,
 "environment": {
 "computeType": "BUILD_GENERAL1_SMALL",
 "image": "aws/codebuild/standard:1.0",
 "type": "LINUX_CONTAINER",
 "environmentVariables": []
 },
 "source": {
 "type": "S3",
 "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
 },
 "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
 "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
 }
 }
}

```

Pour plus d'informations, voir [Créer un projet de construction \(AWS CLI\)](#) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, voir [CreateProject](#) la section Référence des AWS CLI commandes.

## JavaScript

### SDK pour JavaScript (v3)

#### Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Créez un projet.

```

import {
 ArtifactsType,

```

```
CodeBuildClient,
ComputeType,
CreateProjectCommand,
EnvironmentType,
SourceType,
} from "@aws-sdk/client-codebuild";

// Create the AWS CodeBuild project.
export const createProject = async (
 projectName = "MyCodeBuilder",
 roleArn = "arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin",
 buildOutputBucket = "xxxx",
 githubUrl = "https://...",
) => {
 const codeBuildClient = new CodeBuildClient({});

 const response = await codeBuildClient.send(
 new CreateProjectCommand({
 artifacts: {
 // The destination of the build artifacts.
 type: ArtifactsType.S3,
 location: buildOutputBucket,
 },
 // Information about the build environment. The combination of
 "computeType" and "type" determines the
 // requirements for the environment such as CPU, memory, and disk space.
 environment: {
 // Build environment compute types.
 // https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-compute-types.html
 computeType: ComputeType.BUILD_GENERAL1_SMALL,
 // Docker image identifier.
 // See https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-available.html
 image: "aws/codebuild/standard:7.0",
 // Build environment type.
 type: EnvironmentType.LINUX_CONTAINER,
 },
 name: projectName,
 // A role ARN with permission to create a CodeBuild project, write to the
 artifact location, and write CloudWatch logs.
 serviceRole: roleArn,
 source: {
 // The type of repository that contains the source code to be built.
```

```

 type: SourceType.GITHUB,
 // The location of the repository that contains the source code to be
built.
 location: githubUrl,
 },
)),
);
console.log(response);
// {
// '$metadata': {
// httpStatusCode: 200,
// requestId: 'b428b244-777b-49a6-a48d-5dffedced8e7',
// extendedRequestId: undefined,
// cfId: undefined,
// attempts: 1,
// totalRetryDelay: 0
// },
// project: {
// arn: 'arn:aws:codebuild:us-east-1:xxxxxxxxxxxx:project/MyCodeBuilder',
// artifacts: {
// encryptionDisabled: false,
// location: 'xxxxxx-xxxxxxx-xxxxxx',
// name: 'MyCodeBuilder',
// namespaceType: 'NONE',
// packaging: 'NONE',
// type: 'S3'
// },
// badge: { badgeEnabled: false },
// cache: { type: 'NO_CACHE' },
// created: 2023-08-18T14:46:48.979Z,
// encryptionKey: 'arn:aws:kms:us-east-1:xxxxxxxxxxxx:alias/aws/s3',
// environment: {
// computeType: 'BUILD_GENERAL1_SMALL',
// environmentVariables: [],
// image: 'aws/codebuild/standard:7.0',
// imagePullCredentialsType: 'CODEBUILD',
// privilegedMode: false,
// type: 'LINUX_CONTAINER'
// },
// lastModified: 2023-08-18T14:46:48.979Z,
// name: 'MyCodeBuilder',
// projectVisibility: 'PRIVATE',
// queuedTimeoutInMinutes: 480,
// serviceRole: 'arn:aws:iam:xxxxxxxxxxxx:role/CodeBuildAdmin',

```

```
// source: {
// insecureSsl: false,
// location: 'https://...',
// reportBuildStatus: false,
// type: 'GITHUB'
// },
// timeoutInMinutes: 60
// }
// }
return response;
};
```

- Pour plus d'informations, consultez le [Guide du développeur AWS SDK pour JavaScript](#).
- Pour plus de détails sur l'API, voir [CreateProject](#) la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

## Utilisation **ListBuilds** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListBuilds`.

### C++

Kit de développement logiciel (SDK) for C++

#### Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
//! List the CodeBuild builds.
/*!
 \param sortType: 'SortOrderType' type.
 \param clientConfiguration: AWS client configuration.
 \return bool: Function succeeded.
*/
```

```

bool AwsDoc::CodeBuild::listBuilds(Aws::CodeBuild::Model::SortOrderType sortType,
 const Aws::Client::ClientConfiguration
&clientConfiguration) {
 Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

 Aws::CodeBuild::Model::ListBuildsRequest listBuildsRequest;
 listBuildsRequest.SetSortOrder(sortType);

 Aws::String nextToken; // Used for pagination.

 do {
 if (!nextToken.empty()) {
 listBuildsRequest.SetNextToken(nextToken);
 }

 Aws::CodeBuild::Model::ListBuildsOutcome listBuildsOutcome =
codeBuildClient.ListBuilds(
 listBuildsRequest);

 if (listBuildsOutcome.IsSuccess()) {
 const Aws::Vector<Aws::String> &ids =
listBuildsOutcome.GetResult().GetIds();
 if (!ids.empty()) {

 std::cout << "Information about each build:" << std::endl;
 Aws::CodeBuild::Model::BatchGetBuildsRequest getBuildsRequest;
 getBuildsRequest.SetIds(listBuildsOutcome.GetResult().GetIds());
 Aws::CodeBuild::Model::BatchGetBuildsOutcome getBuildsOutcome =
codeBuildClient.BatchGetBuilds(
 getBuildsRequest);

 if (getBuildsOutcome.IsSuccess()) {
 const Aws::Vector<Aws::CodeBuild::Model::Build> &builds =
getBuildsOutcome.GetResult().GetBuilds();
 std::cout << builds.size() << " build(s) found." <<
std::endl;

 for (auto val: builds) {
 std::cout << val.GetId() << std::endl;
 }
 } else {
 std::cerr << "Error getting builds"
 << getBuildsOutcome.GetError().GetMessage() <<
std::endl;

 return false;
 }
 }
 }
 } while (listBuildsOutcome.IsTruncated());
}

```

```
 }
 } else {
 std::cout << "No builds found." << std::endl;
 }

 // Get the next token for pagination.

 nextToken = listBuildsOutcome.GetResult().GetNextToken();
} else {
 std::cerr << "Error listing builds"
 << listBuildsOutcome.GetError().GetMessage()
 << std::endl;
 return false;
}

} while (!nextToken.

 empty()

);

 return true;
}
```

- Pour plus de détails sur l'API, voir [ListBuilds](#) la section Référence des AWS SDK pour C++ API.

## CLI

### AWS CLI

Pour obtenir une liste des AWS CodeBuild versions IDs.

L'`list-builds`exemple suivant obtient une liste de CodeBuild IDs triés par ordre croissant.

```
aws codebuild list-builds --sort-order ASCENDING
```

La sortie inclut une valeur `nextToken` qui indique qu'il y a plus de sorties disponibles.

```
{
```



```

 "nextToken": "4AEA6u7J...The full token has been omitted for
 brevity...MzY20A==",
 "ids": [
 "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
 "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}

```

Exécutez à nouveau cette commande et fournissez la valeur `nextToken` de la réponse précédente en tant que paramètre pour obtenir la partie suivante de la sortie. Répétez l'opération jusqu'à ce que vous ne receviez plus de valeur `nextToken` dans la réponse.

```

aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==

```

Partie suivante de la sortie :

```

{
 "ids": [
 "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
 "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
]
}

```

Pour plus d'informations, voir [Afficher une liste de versions IDs \(AWS CLI\)](#) dans le guide de AWS CodeBuild l'utilisateur

- Pour plus de détails sur l'API, voir [ListBuilds](#) la section Référence des AWS CLI commandes.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

## Utilisation **ListProjects** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListProjects`.

## C++

## Kit de développement logiciel (SDK) for C++

 Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
//! List the CodeBuild projects.
/*!
 \param sortType: 'SortOrderType' type.
 \param clientConfiguration: AWS client configuration.
 \return bool: Function succeeded.
*/
bool AwsDoc::CodeBuild::listProjects(Aws::CodeBuild::Model::SortOrderType
 sortType,
 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
 Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

 Aws::CodeBuild::Model::ListProjectsRequest listProjectsRequest;
 listProjectsRequest.SetSortOrder(sortType);

 Aws::String nextToken; // Next token for pagination.
 Aws::Vector<Aws::String> allProjects;

 do {
 if (!nextToken.empty()) {
 listProjectsRequest.SetNextToken(nextToken);
 }

 Aws::CodeBuild::Model::ListProjectsOutcome outcome =
 codeBuildClient.ListProjects(
 listProjectsRequest);

 if (outcome.IsSuccess()) {
 const Aws::Vector<Aws::String> &projects =
 outcome.GetResult().GetProjects();
 allProjects.insert(allProjects.end(), projects.begin(),
 projects.end());
 }
 } while (outcome.IsSuccess() && !nextToken.empty());
}
```

```

 nextToken = outcome.GetResult().GetNextToken();
 }

 else {
 std::cerr << "Error listing projects" <<
outcome.GetError().GetMessage()
 << std::endl;
 }

 } while (!nextToken.empty());

 std::cout << allProjects.size() << " project(s) found." << std::endl;
 for (auto project: allProjects) {
 std::cout << project << std::endl;
 }

 return true;
}

```

- Pour plus de détails sur l'API, voir [ListProjects](#) la section Référence des AWS SDK pour C++ API.

## CLI

### AWS CLI

Pour obtenir une liste des noms de projets de AWS CodeBuild construction.

L'`list-projectsexemple` suivant obtient une liste de projets de CodeBuild construction triés par nom dans l'ordre croissant.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

La sortie inclut une valeur `nextToken` qui indique qu'il y a plus de sorties disponibles.

```

{
 "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
 "projects": [
 "codebuild-demo-project",

```

```

 "codebuild-demo-project2",
 ... The full list of build project names has been omitted for
 brevity ...
 "codebuild-demo-project99"
]
}

```

Exécutez à nouveau cette commande et fournissez la valeur `nextToken` de la réponse précédente en tant que paramètre pour obtenir la partie suivante de la sortie. Répétez l'opération jusqu'à ce que vous ne receviez plus de valeur `nextToken` dans la réponse.

```

aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
 "projects": [
 "codebuild-demo-project100",
 "codebuild-demo-project101",

 ... The full list of build project names has been omitted for brevity ...
 "codebuild-demo-project122"
]
}

```

Pour plus d'informations, voir [Afficher une liste de noms de projets de construction \(AWS CLI\)](#) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, voir [ListProjects](#) la section Référence des AWS CLI commandes.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

## Utilisation **StartBuild** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `StartBuild`.

## C++

## Kit de développement logiciel (SDK) for C++

 Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
#!/ Start an AWS CodeBuild project build.
/*!
 \param projectName: A CodeBuild project name.
 \param clientConfiguration: AWS client configuration.
 \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::startBuild(const Aws::String &projectName,
 const Aws::Client::ClientConfiguration
&clientConfiguration) {
 Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

 Aws::CodeBuild::Model::StartBuildRequest startBuildRequest;
 startBuildRequest.SetProjectName(projectName);

 Aws::CodeBuild::Model::StartBuildOutcome outcome =
codeBuildClient.StartBuild(
 startBuildRequest);

 if (outcome.IsSuccess()) {
 std::cout << "Successfully started build" << std::endl;
 std::cout << "Build ID: " << outcome.GetResult().GetBuild().GetId()
 << std::endl;
 }

 else {
 std::cerr << "Error starting build" << outcome.GetError().GetMessage()
 << std::endl;
 }

 return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, voir [StartBuild](#) la section Référence des AWS SDK pour C++ API.

## CLI

### AWS CLI

Pour commencer à exécuter la compilation d'un projet de AWS CodeBuild construction.

L'`start-build`exemple suivant lance une compilation pour le CodeBuild projet spécifié. La génération remplace à la fois le paramètre du projet concernant le nombre de minutes que la génération est autorisée à être mise en file d'attente avant son expiration et les paramètres d'artefact du projet.

```
aws codebuild start-build \
 --project-name "my-demo-project" \
 --queued-timeout-in-minutes-override 5 \
 --artifacts-override {"type": "S3","location":
 \"arn:aws:s3::artifacts-override\", \"overrideArtifactName\": true}
```

Sortie :

```
{
 "build": {
 "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
 "buildStatus": "IN_PROGRESS",
 "buildComplete": false,
 "projectName": "my-demo-project",
 "timeoutInMinutes": 60,
 "source": {
 "insecureSsl": false,
 "type": "S3",
 "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip"
 },
 "queuedTimeoutInMinutes": 5,
 "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
 "currentPhase": "QUEUED",
 "startTime": 1556905683.568,
 "environment": {
```

```

 "computeType": "BUILD_GENERAL1_MEDIUM",
 "environmentVariables": [],
 "type": "LINUX_CONTAINER",
 "privilegedMode": false,
 "image": "aws/codebuild/standard:1.0",
 "imagePullCredentialsType": "CODEBUILD"
 },
 "phases": [
 {
 "phaseStatus": "SUCCEEDED",
 "startTime": 1556905683.568,
 "phaseType": "SUBMITTED",
 "durationInSeconds": 0,
 "endTime": 1556905684.524
 },
 {
 "startTime": 1556905684.524,
 "phaseType": "QUEUED"
 }
],
 "logs": {
 "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=null;stream=null"
 },
 "artifacts": {
 "encryptionDisabled": false,
 "location": "arn:aws:s3:::artifacts-override/my-demo-project",
 "overrideArtifactName": true
 },
 "cache": {
 "type": "NO_CACHE"
 },
 "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
 "initiator": "my-aws-account-name",
 "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
 }
}

```

Pour plus d'informations, voir [Run a Build \(AWS CLI\)](#) dans le guide de AWS CodeBuild l'utilisateur.

- Pour plus de détails sur l'API, voir [StartBuild](#) la section Référence des AWS CLI commandes.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.



# Résolution des problèmes AWS CodeBuild

Utilisez les informations de cette rubrique pour identifier, diagnostiquer et résoudre des problèmes. Pour savoir comment enregistrer et surveiller les CodeBuild versions afin de résoudre les problèmes, consultez [Journalisation et surveillance](#).

## Rubriques

- [Apache Maven génère des artefacts de référence à partir du mauvais répertoire](#)
- [Les commandes de génération s'exécutent en tant que racine par défaut](#)
- [Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains. Caractères anglais](#)
- [Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store](#)
- [Impossible d'accéder au filtre de branche dans la CodeBuild console](#)
- [Impossible d'afficher la réussite ou l'échec de la génération](#)
- [État du build non communiqué au fournisseur source](#)
- [Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019](#)
- [Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures](#)
- [Erreur : accès refusé lors de la tentative de téléchargement du cache](#)
- [Erreur : « BUILD\\_CONTAINER\\_UNABLE\\_TO\\_PULL\\_IMAGE » lors de l'utilisation d'une image de génération personnalisée](#)
- [Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces](#)
- [Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération](#)
- [Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction](#)
- [Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl »](#)
- [Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération](#)

- [Erreur : « Git clone failed: Unable to access 'your-repository-URL': SSL certificate problem: Self signed certificate »](#)
- [Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération](#)
- [Erreur : « This build image requires selecting at least one runtime version. »](#)
- [Erreur : « QUEUED: INSUFFICIENT\\_SUBNET » lorsqu'une génération dans une file d'attente échoue](#)
- [Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie »](#)
- [Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied»](#)
- [Erreur : « Unable to locate credentials »](#)
- [RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy](#)
- [Le shell bourne \(sh\) doit exister dans les images de génération](#)
- [Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération](#)
- [Erreur : « Impossible de vérifier l' JobWorker identité » lors de l'ouverture de la CodeBuild console](#)
- [La construction n'a pas pu démarrer](#)
- [Accès aux GitHub métadonnées dans les versions mises en cache localement](#)
- [AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3...](#)
- [Erreur : « Vos informations d'identification ne disposent pas d'une ou de plusieurs étendues de privilèges requises » lors de la création d'un CodeBuild projet avec CodeConnections](#)
- [Erreur : « Désolé, aucun terminal n'a été demandé, impossible d'obtenir des informations » lors de la compilation avec la commande d'installation d'Ubuntu](#)

## Apache Maven génère des artefacts de référence à partir du mauvais répertoire

Problème : [Lorsque vous utilisez Maven avec un environnement de génération Java AWS CodeBuild fourni, Maven extrait les dépendances de build et de plugin du référentiel central sécurisé de Maven à l'adresse <https://repo1.maven.org/maven2>. Cela se produit même si le fichier pom.xml de votre projet de génération déclare explicitement d'autres emplacements à utiliser.](#)

Cause possible : les environnements de génération Java CodeBuild fournis incluent un fichier nommé `settings.xml` qui est préinstallé dans le répertoire de l'environnement de `/root/.m2` construction. Ce fichier `settings.xml` contient les déclarations suivantes qui demandent à Maven de toujours extraire les dépendances de génération et de plug-in du référentiel Maven central sécurisé situé à l'adresse <https://repo1.maven.org/maven2>.

```
<settings>
 <activeProfiles>
 <activeProfile>securecentral</activeProfile>
 </activeProfiles>
 <profiles>
 <profile>
 <id>securecentral</id>
 <repositories>
 <repository>
 <id>central</id>
 <url>https://repo1.maven.org/maven2</url>
 <releases>
 <enabled>true</enabled>
 </releases>
 </repository>
 </repositories>
 <pluginRepositories>
 <pluginRepository>
 <id>central</id>
 <url>https://repo1.maven.org/maven2</url>
 <releases>
 <enabled>true</enabled>
 </releases>
 </pluginRepository>
 </pluginRepositories>
 </profile>
 </profiles>
</settings>
```

Solution recommandée : effectuez les opérations suivantes :

1. Ajoutez un fichier `settings.xml` à votre code source.
2. Dans ce fichier `settings.xml`, utilisez le format de fichier `settings.xml` précédent comme guide pour déclarer les référentiels desquels vous voulez que Maven extraie plutôt les dépendances de génération et de plug-in.

3. Au cours de la `install` phase de votre projet de construction, demandez CodeBuild de copier votre `settings.xml` fichier dans le `/root/.m2` répertoire de l'environnement de construction. Par exemple, examinez l'extrait de code suivant d'un fichier `buildspec.yml` qui illustre ce comportement.

```
version 0.2

phases:
 install:
 commands:
 - cp ./settings.xml /root/.m2/settings.xml
```

## Les commandes de génération s'exécutent en tant que racine par défaut

**Problème :** AWS CodeBuild exécute vos commandes de compilation en tant qu'utilisateur `root`. Cela se produit même si votre Dockerfile d'image de génération associé définit l'instruction `USER` sur un autre utilisateur.

**Cause :** Par défaut, CodeBuild exécute toutes les commandes de compilation en tant qu'utilisateur `root`.

**Solution recommandée :** aucune.

## Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains. Caractères anglais

**Problème :** lorsque vous exécutez une version qui utilise des fichiers dont le nom contient des informations non américaines Caractères anglais (par exemple, caractères chinois), la compilation échoue.

**Cause possible :** les environnements de construction fournis par AWS CodeBuild ont leurs paramètres régionaux par défaut définis sur `POSIX`. `POSIX` les paramètres de localisation sont moins compatibles avec les noms de fichiers non américains CodeBuild et les noms de fichiers contenant des informations non américaines Caractères anglais et peuvent entraîner l'échec des versions associées.

Solution recommandée : ajoutez les commandes suivantes à la section `pre_build` de votre fichier `buildspec`. Ces commandes font en sorte que l'environnement de construction utilise l'UTF-8 en anglais américain pour ses paramètres de localisation, ce qui est plus compatible avec CodeBuild les noms de fichiers contenant des informations non américaines. Caractères anglais.

Pour les environnements de génération basés sur Ubuntu :

```
pre_build:
 commands:
 - export LC_ALL="en_US.UTF-8"
 - locale-gen en_US en_US.UTF-8
 - dpkg-reconfigure -f noninteractive locales
```

Pour les environnements de création basés sur Amazon Linux :

```
pre_build:
 commands:
 - export LC_ALL="en_US.utf8"
```

## Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store

Problème : lorsqu'un build essaie d'obtenir la valeur d'un ou de plusieurs paramètres stockés dans le magasin de paramètres Amazon EC2, le build échoue dans la `DOWNLOAD_SOURCE` phase d'erreur. `Parameter does not exist`

Cause possible : le rôle de service sur lequel repose le projet de génération n'est pas autorisé à appeler `ssm:GetParameters` ou le projet de génération utilise un rôle de service généré par AWS CodeBuild et permettant d'appeler `ssm:GetParameters`, mais les paramètres ont des noms qui ne commencent pas par `/CodeBuild/`.

Solutions recommandées :

- Si le rôle de service n'a pas été généré par CodeBuild, mettez à jour sa définition CodeBuild pour permettre d'appeler `ssm:GetParameters`. Par exemple, la déclaration de stratégie suivante permet d'appeler l'action `ssm:GetParameters` pour obtenir des paramètres avec des noms commençant par `/CodeBuild/` :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "ssm:GetParameters",
 "Effect": "Allow",
 "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
CodeBuild/*"
 }
]
}
```

- Si le rôle de service a été généré par CodeBuild, mettez à jour sa définition pour autoriser l'accès CodeBuild aux paramètres dans Amazon EC2 Parameter Store avec des noms autres que ceux commençant par. /CodeBuild/ Par exemple, la déclaration de stratégie suivante permet d'appeler l'action `ssm:GetParameters` pour obtenir des paramètres avec le nom spécifié :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "ssm:GetParameters",
 "Effect": "Allow",
 "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/PARAMETER_NAME"
 }
]
}
```

## Impossible d'accéder au filtre de branche dans la CodeBuild console

Problème : L'option de filtre de branche n'est pas disponible dans la console lorsque vous créez ou mettez à jour un AWS CodeBuild projet.

Cause possible : l'option de filtre de branche est obsolète. Elle a été remplacée par les groupes de filtres webhook, qui fournissent davantage de contrôle sur les événements webhook qui déclenchent une nouvelle génération CodeBuild.

Solution recommandée : pour migrer un filtre de branche créé avant l'introduction des filtres webhook, créez un groupes de filtres webhook avec un filtre HEAD\_REF avec l'expression régulière `^refs/heads/branchName$`. Par exemple, si votre expression régulière de filtre de branche était `^branchName$`, l'expression régulière mise à jour que vous placez dans le filtre HEAD\_REF est `^refs/heads/branchName$`. Pour plus d'informations, consultez [Événements du webhook Bitbucket](#) et [Filtrer les événements du GitHub webhook \(console\)](#).

## Impossible d'afficher la réussite ou l'échec de la génération

Problème : vous ne pouvez pas consulter l'état de réussite ou d'échec lorsqu'une génération a fait l'objet d'une nouvelle tentative.

Cause possible : l'option qui permet de signaler le statut de votre génération n'est pas activée.

Solutions recommandées : Activez le statut de génération du rapport lorsque vous créez ou mettez à jour un CodeBuild projet. Cette option indique à CodeBuild de renvoyer l'état lorsque vous déclenchez une génération. Pour plus d'informations, consultez [reportBuildStatus](#) dans la Référence d'API AWS CodeBuild .

## État du build non communiqué au fournisseur source

Problème : Après avoir autorisé le signalement de l'état du build à un fournisseur source, tel que GitHub Bitbucket, le statut du build n'est pas mis à jour.

Cause possible : L'utilisateur associé au fournisseur de source n'a pas accès en écriture au dépôt.

Solutions recommandées : Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

# Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019

Problème : Impossible de trouver ou de sélectionner l'image de base de la plateforme Windows Server Core 2019.

Cause possible : vous utilisez une AWS région qui ne prend pas en charge cette image.

Solutions recommandées : utilisez l'une des AWS régions suivantes où l'image de base de la plateforme Windows Server Core 2019 est prise en charge :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Oregon)
- Europe (Irlande)

## Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures

Problème : les résultats d'une ou plusieurs commandes dans votre fichier buildspec ne sont pas reconnus par des commandes ultérieures dans le même fichier buildspec. Par exemple, une commande peut définir une variable d'environnement local, mais une commande exécutée ultérieurement ne parvient pas à obtenir la valeur de cette variable d'environnement local.

Cause possible : dans le fichier buildspec version 0.1, AWS CodeBuild exécute chaque commande dans une instance distincte du shell par défaut dans l'environnement de génération. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes.

Solutions recommandées : nous vous recommandons d'utiliser la version 0.2 de buildspec afin de résoudre ce problème. Si vous devez utiliser la version 0.1 de buildspec pour une raison quelconque, nous vous recommandons d'exploiter l'opérateur de chaînage de commande shell (par exemple, && dans Linux) pour combiner plusieurs commandes en une seule. Ou incluez dans votre code source un script shell contenant plusieurs commandes, puis appelez ce script shell à partir d'une commande unique dans le fichier buildspec. Pour plus d'informations, consultez [Shells et commandes dans les environnements de génération](#) et [Variables d'environnement dans les environnements de génération](#).



## Erreur : accès refusé lors de la tentative de téléchargement du cache

Problème : lors de la tentative de téléchargement du cache sur un projet de génération dont le cache est activé, vous recevez l'erreur `Access denied`.

Causes possibles :

- vous venez de configurer la mise en cache dans le cadre de votre projet de génération.
- Le cache a récemment été invalidé par le biais de l'API `InvalidateProjectCache`.
- Le rôle de service utilisé par CodeBuild n'a aucune `s3:PutObject` autorisation `s3:GetObject` d'accès au compartiment S3 qui contient le cache.

Solution recommandée : pour la première utilisation, il est normal de voir cela immédiatement après la mise à jour de la configuration du cache. Si cette erreur persiste, vérifiez si le rôle de service a les autorisations `s3:GetObject` et `s3:PutObject` pour le compartiment S3 qui contient le cache. Pour plus d'informations, consultez [Spécifier les autorisations S3](#) dans le manuel Amazon S3 Developer Guide.

## Erreur : « BUILD\_CONTAINER\_UNABLE\_TO\_PULL\_IMAGE » lors de l'utilisation d'une image de génération personnalisée

Problème : lorsque vous essayez d'exécuter une génération qui utilise une version de génération personnalisée, la génération échoue avec l'erreur `BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE`.

Cause possible : la taille globale non compressée de l'image de génération est supérieure à l'espace disque disponible du type de calcul de l'environnement de génération. Pour vérifier la taille de votre image de génération, utilisez Docker pour exécuter la commande `docker images REPOSITORY:TAG`. Pour obtenir une liste de l'espace disque disponible par type de calcul, reportez-vous à la section [Modes et types de calcul de l'environnement de création](#).

Solution recommandée : utilisez un type de calcul plus important avec plus d'espace disque disponible, ou réduisez la taille de votre image de construction personnalisée.

Cause possible : AWS CodeBuild n'est pas autorisé à extraire l'image de compilation de votre Amazon Elastic Container Registry (Amazon ECR).

Solution recommandée : mettez à jour les autorisations de votre référentiel dans Amazon ECR afin de CodeBuild pouvoir intégrer votre image de construction personnalisée dans l'environnement de création. Pour de plus amples informations, veuillez consulter [Exemple Amazon ECR](#).

Cause possible : L'image Amazon ECR que vous avez demandée n'est pas disponible dans la AWS région utilisée par votre AWS compte.

Solution recommandée : utilisez une image Amazon ECR située dans la même AWS région que celle utilisée par votre AWS compte.

Cause possible : vous utilisez un registre privé dans un VPC qui ne dispose pas d'un accès public à Internet. CodeBuild Impossible d'extraire une image d'une adresse IP privée dans un VPC. Pour de plus amples informations, veuillez consulter [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#).

Solution recommandée : Si vous utilisez un registre privé dans un VPC, assurez-vous que celui-ci dispose d'un accès public à Internet.

Cause possible : si le message d'erreur contient `toomanyrequests` et que l'image provient de Docker Hub, cette erreur signifie que la limite d'attraction de Docker Hub a été atteinte.

Solution recommandée : utilisez un registre privé Docker Hub ou procurez-vous votre image auprès d'Amazon ECR. Pour plus d'informations sur l'utilisation d'un registre privé, consultez [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#). Pour plus d'informations sur l'utilisation d'Amazon ECR, consultez [Exemple Amazon ECR pour CodeBuild](#).

**Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces**

Problème : Lorsque vous essayez d'utiliser un conteneur Microsoft Windows ou Linux dans AWS CodeBuild, cette erreur se produit pendant la phase de PROVISIONING.

Causes possibles :

- La version du système d'exploitation du conteneur n'est pas prise en charge par CodeBuild.
- HTTP\_PROXY, HTTPS\_PROXY, ou les deux sont spécifiés dans le conteneur.

Solutions recommandées :

- Pour Microsoft Windows, utilisez un conteneur Windows avec un système d'exploitation de conteneur de version : microsoft/windowsservercore:10.0.x (for example, microsoft/windowsservercore 10.0.14393.2125).
- Pour Linux, effacez les paramètres HTTP\_PROXY et HTTPS\_PROXY de votre image Docker, ou spécifiez la configuration du VPC dans votre projet.

## Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération

Problème : la génération échoue et vous recevez une erreur similaire à `Cannot connect to the Docker daemon at unix:/var/run/docker.sock. Is the docker daemon running?` dans le journal de génération.

Cause possible : vous n'exécutez pas la génération en mode privilégié.

Solution recommandée : pour corriger cette erreur, vous devez activer le mode privilégié et mettre à jour votre buildspec en suivant les instructions suivantes.

Pour exécuter votre build en mode privilégié, procédez comme suit :

1. Ouvrez la CodeBuild console à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans le volet de navigation, choisissez Build projects, puis choisissez votre projet de build.
3. Dans Modifier, choisissez Environnement.
4. Sélectionnez Additional configuration (Configuration supplémentaire).
5. Dans Privileged, sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos builds bénéficient de privilèges élevés. .
6. Choisissez Mettre à jour l'environnement.
7. Choisissez Démarrer la génération pour réessayer de créer votre génération.

Vous devrez également démarrer le démon Docker dans votre conteneur. La `install` phase de votre `buildspec` peut ressembler à ceci.

```
phases:
 install:
 commands:
 - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
 host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
 - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Pour plus d'informations sur le pilote de stockage OverlayFS référencé dans le fichier `buildspec`, consultez [Utilisation du pilote de stockage OverlayFS](#) sur le site web Docker.

#### Note

Si le système d'exploitation de base est Alpine Linux, dans `buildspec.yml` ajoutez l'argument `-t` à `timeout` :

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Pour en savoir plus sur la création et l'exécution d'une image Docker à l'aide AWS CodeBuild de [Docker dans un exemple d'image personnalisé pour CodeBuild](#)

## Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction

Problème : lorsque vous essayez de créer ou de mettre à jour un projet de génération, vous recevez l'erreur `Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name`.

Causes possibles :

- Le AWS Security Token Service (AWS STS) a été désactivé pour la AWS région dans laquelle vous essayez de créer ou de mettre à jour le projet de construction.
- Le rôle AWS CodeBuild de service associé au projet de construction n'existe pas ou ne dispose pas d'autorisations suffisantes pour être approuvé CodeBuild.

- Le boîtier AWS CodeBuild de rôle de service associé au projet de construction ne correspond pas au rôle IAM réel.

#### Solutions recommandées :

- Assurez-vous qu' AWS STS il est activé pour la AWS région dans laquelle vous essayez de créer ou de mettre à jour le projet de construction. Pour plus d'informations, consultez la section [Activation et désactivation AWS STS dans une AWS région](#) dans le guide de l'utilisateur IAM.
- Assurez-vous que le rôle CodeBuild de service cible existe dans votre AWS compte. Si vous n'utilisez pas la console, vérifiez que vous avez bien orthographié l'Amazon Resource Name (ARN) du rôle de service lorsque vous avez créé ou mis à jour le projet de génération. Notez que les rôles IAM distinguent les majuscules et minuscules. Vérifiez donc que le boîtier du rôle IAM est correct.
- Assurez-vous que le rôle de CodeBuild service cible dispose d'autorisations suffisantes pour être fiable CodeBuild. Pour plus d'informations, consultez la déclaration de stratégie de relation d'approbation dans [CodeBuild Autoriser l'interaction avec d'autres AWS services](#).

**Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl »**

**Problème :** lorsque vous exécutez une génération, vous recevez une erreur sur une modification de la propriété d'un compartiment S3 et d'autorisations GetBucketAcl.

**Cause possible :** vous avez ajouté les `s3:GetBucketLocation` autorisations `s3:GetBucketAcl` et à votre rôle IAM. Ces autorisations protègent le compartiment S3 de votre projet et garantissent que vous seul pouvez y accéder. Après l'ajout de ces autorisations, le propriétaire du compartiment S3 a changé.

**Solution recommandée :** vérifiez que vous êtes propriétaire du compartiment S3, puis ajoutez de nouveau des autorisations à votre rôle IAM. Pour de plus amples informations, veuillez consulter [Accès sécurisé aux compartiments S3](#).

## Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération

Problème : lorsque vous exécutez une génération, la phase `UPLOAD_ARTIFACTS` de la génération échoue avec l'erreur `Failed to upload artifacts: Invalid arn`.

Cause possible : votre compartiment de sortie S3 (le compartiment AWS CodeBuild dans lequel sont stockés les résultats de la génération) se trouve dans une AWS région différente de celle du projet de CodeBuild construction.

Solution recommandée : mettez à jour les paramètres du projet de génération pour qu'ils pointent vers un compartiment de sortie situé dans la même AWS région que le projet de construction.

## Erreur : « Git clone failed: Unable to access '**your-repository-URL**': SSL certificate problem: Self signed certificate »

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Cause possible : Votre référentiel source possède un certificat auto-signé, mais vous n'avez pas choisi d'installer le certificat depuis votre compartiment S3 dans le cadre de votre projet de génération.

Solutions recommandées :

- modifiez votre projet. Pour Certificat, choisissez Installer le certificat à partir de votre compartiment S3. Pour Compartiment de certificat, choisissez le compartiment S3 dans lequel votre certificat SSL est stocké. Pour Object key of certificate (Clé d'objet de certificat), tapez le nom de la clé d'objet S3.
- modifiez votre projet. Sélectionnez SSL non sécurisé pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub Enterprise Server.

### Note

Nous vous recommandons de n'utiliser Insecure SSL que pour les tests. Cette option ne doit pas être utilisée dans un environnement de production.

## Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération

**Problème :** lorsque vous exécutez une génération, la phase `DOWNLOAD_SOURCE` de la génération échoue avec l'erreur `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

**Cause possible :** votre code source prédéfini est stocké dans un compartiment S3, et ce compartiment se trouve dans une AWS région différente de celle du projet de AWS CodeBuild génération.

**Solution recommandée :** mettez à jour les paramètres du projet de génération pour pointer vers un compartiment qui contient votre code source précompilé. Assurez-vous que le bucket se trouve dans la même AWS région que le projet de construction.

## Erreur : « This build image requires selecting at least one runtime version. »

**Problème :** lorsque vous exécutez une génération, la phase `DOWNLOAD_SOURCE` de la génération échoue avec l'erreur `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

**Cause possible :** votre build utilise la version 1.0 ou ultérieure de l'image standard Amazon Linux 2 (AL2), ou la version 2.0 ou ultérieure de l'image standard Ubuntu, et aucun environnement d'exécution n'est spécifié dans le fichier `buildspec`.

**Solution recommandée :** Si vous utilisez l'image `aws/codebuild/standard:2.0` CodeBuild gérée, vous devez spécifier une version d'exécution dans la `runtime-versions` section du fichier `buildspec`. Par exemple, vous pouvez utiliser le fichier `buildspec` suivant pour un projet qui utilise PHP :

```
version: 0.2

phases:
 install:
 runtime-versions:
```

```
 php: 7.3
 build:
 commands:
 - php --version
 artifacts:
 files:
 - README.md
```

#### Note

Si vous spécifiez une `runtime-versions` section et utilisez une image autre qu'Ubuntu Standard Image 2.0 ou version ultérieure, ou l'image standard Amazon Linux 2 (AL2) 1.0 ou version ultérieure, le build émet l'avertissement « `Skipping install of runtimes. Runtime version selection is not supported by this build image.` ».

Pour de plus amples informations, veuillez consulter [Specify runtime versions in the buildspec file](#).

## Erreur : « QUEUED: INSUFFICIENT\_SUBNET » lorsqu'une génération dans une file d'attente échoue

Problème : une génération dans une file d'attente échoue avec une erreur similaire à `QUEUED: INSUFFICIENT_SUBNET`.

Causes possibles : Le bloc IPv4 CIDR spécifié pour votre VPC utilise une adresse IP réservée. Les quatre premières adresses IP et la dernière adresse IP du bloc d'adresse CIDR de chaque sous-réseau ne sont pas disponibles pour utilisation, et ne peuvent donc pas être affectées à une instance. Par exemple, dans un sous-réseau avec le bloc d'adresse CIDR `10.0.0.0/24`, les cinq adresses IP suivantes sont réservées :

- `10.0.0.0` : Adresse réseau.
- `10.0.0.1` : Réserve par AWS pour le routeur VPC.
- `10.0.0.2` : Réserve par AWS. L'adresse IP du serveur DNS est toujours la base de la plage d'adresses du réseau VPC, plus deux ; cependant, nous réservons également la base de chaque sous-réseau, plus deux. Dans le VPCs cas de plusieurs blocs d'adresse CIDR, l'adresse IP du serveur DNS se trouve dans le CIDR principal. Pour plus d'informations, consultez [Serveur Amazon DNS](#) dans le Guide de l'utilisateur Amazon VPC.
- `10.0.0.3` : Réserve par AWS pour une utilisation future.



- 10.0.0.255 : Adresse de diffusion réseau. Nous ne prenons pas en charge la diffusion dans un VPC. Cette adresse est réservée.

Solutions recommandées : vérifiez si votre VPC utilise une adresse IP réservée. Remplacez toute adresse IP réservée par une adresse non réservée. Pour plus d'informations, consultez [Dimensionnement des VPC et des sous-réseaux](#) dans le Amazon VPC Guide de l'utilisateur.

**Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie »**

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Cause possible : vous avez configuré la mise en cache dans le cadre de votre projet de génération et vous utilisez une ancienne image Docker qui inclut un certificat racine expiré.

Solution recommandée : mettez à jour l'image Docker utilisée dans votre AWS CodeBuild projet. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).

**Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied»**

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Causes possibles :

- vous avez choisi le mauvais compartiment S3 pour votre certificat.
- Vous avez entré la mauvaise clé d'objet pour votre certificat.

Solutions recommandées :

- modifiez votre projet. Pour Compartiment de certificat, choisissez le compartiment S3 dans lequel votre certificat SSL est stocké.

- modifiez votre projet. Pour Object key of certificate (Clé d'objet de certificat), tapez le nom de la clé d'objet S3.

## Erreur : « Unable to locate credentials »

Problème : Lorsque vous essayez d'exécuter le AWS CLI, d'utiliser un AWS SDK ou d'appeler un autre composant similaire dans le cadre d'une compilation, vous obtenez des erreurs de génération directement liées au AWS CLI AWS SDK ou au composant. Par exemple, vous pouvez obtenir l'erreur de génération `Unable to locate credentials`.

Causes possibles :

- La version du AWS CLI AWS SDK ou du composant de l'environnement de construction est incompatible avec AWS CodeBuild.
- Vous exécutez un conteneur Docker dans un environnement de génération qui utilise Docker, et le conteneur n'a pas accès aux AWS informations d'identification par défaut.

Solutions recommandées :

- Assurez-vous que votre environnement de génération dispose de la version suivante ou supérieure du AWS CLI AWS SDK ou du composant.
  - AWS CLI : 1.10.47
  - AWS SDK pour C++ : 0.2.19
  - AWS SDK pour Go : 1.2.5
  - AWS SDK pour Java : 1.11.16
  - AWS SDK pour JavaScript : 2.4.7
  - AWS SDK pour PHP : 3.18.28
  - AWS SDK pour Python (Boto3) : 1.4.0
  - AWS SDK pour Ruby : 2.3.22
  - Botocore : 1.4.37
  - CoreCLR : 3.2.6-beta
  - Node.js : 2.4.7
- Si vous devez exécuter un conteneur Docker dans un environnement de génération et que le conteneur nécessite des AWS informations d'identification, vous devez transmettre les informations

d'identification de l'environnement de génération au conteneur. Dans votre fichier buildspec, incluez une commande Docker `run` telle que la suivante. Cet exemple utilise la commande `aws s3 ls` pour répertorier les compartiments S3 disponibles. L'option passe par les variables d'environnement requises pour que votre conteneur puisse accéder aux AWS informations d'identification.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Si vous créez une image Docker et que la compilation nécessite des AWS informations d'identification (par exemple, pour télécharger un fichier depuis Amazon S3), vous devez transmettre les informations d'identification de l'environnement de génération au processus de génération Docker comme suit.

1. Dans votre Dockerfile de code source pour l'image Docker, spécifiez les instructions ARG suivantes.

```
ARG AWS_DEFAULT_REGION
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. Dans votre fichier buildspec, incluez une commande Docker `build` telle que la suivante. Les `--build-arg` options définissent les variables d'environnement requises pour que votre processus de création de Docker puisse accéder aux AWS informations d'identification.

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -
t your-image-tag .
```

## RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy

Problème : Vous recevez une erreur `RequestError` similaire à l'une des erreurs suivantes :

- `RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeout` de CloudWatch Logs.

- Error uploading artifacts: RequestError: send request failed caused by: Put https://*your-bucket*.s3.*your-aws-region*.amazonaws.com/\*: dial tcp 52.219.96.208:443: connect: connection refused depuis Amazon S3.

Causes possibles :

- `ssl-bump` n'est pas configuré correctement.
- La stratégie de sécurité de votre organisation ne vous permet pas d'utiliser `ssl_bump`.
- Votre fichier `buildspec` n'a pas de paramètres proxy spécifiés à l'aide d'un élément `proxy`.

Solutions recommandées :

- assurez-vous que `ssl-bump` est configuré correctement. Si vous utilisez Squid pour votre serveur proxy, consultez [Configuration de Squid en tant que serveur proxy explicite](#).
- Pour utiliser des points de terminaison privés pour Amazon S3 et CloudWatch Logs, procédez comme suit :
  1. Dans la table de routage de votre sous-réseau privé, supprimez la règle que vous avez ajoutée qui achemine le trafic destiné à Internet vers votre serveur proxy. Pour plus d'informations, consultez [la section Création d'un sous-réseau dans votre VPC](#) dans le guide de l'utilisateur Amazon VPC.
  2. Créez un point de terminaison Amazon S3 et un point de terminaison CloudWatch Logs privés et associez-les au sous-réseau privé de votre Amazon VPC. Pour plus d'informations, consultez la section [Services de point de terminaison VPC](#) dans le guide de l'utilisateur Amazon VPC.
  3. Vérifiez que l'option Activer le nom DNS privé dans votre Amazon VPC est sélectionnée. Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.
- Si vous n'utilisez pas `ssl-bump` pour un serveur proxy explicite, ajoutez une configuration proxy à votre fichier `buildspec` à l'aide d'un élément `proxy`. Pour plus d'informations, consultez [Exécuter CodeBuild sur un serveur proxy explicite](#) et [Syntaxe d'un fichier buildspec](#).

```
version: 0.2
proxy:
 upload-artifacts: yes
 logs: yes
```

```
phases:
 build:
 commands:
```

## Le shell bourne (sh) doit exister dans les images de génération

Problème : vous utilisez une image de build qui n'est pas fournie par AWS CodeBuild, et vos builds échouent avec le messageBuild `container found dead before completing the build`.

Cause possible : Le shell Bourne (sh) n'est pas inclus dans votre image de construction. CodeBuild doit sh exécuter des commandes et des scripts de construction.

Solution recommandée : S'il sh n'est pas présent dans votre image de construction, assurez-vous de l'inclure avant de commencer toute autre génération utilisant votre image. (inclut CodeBuild déjà des images sh dans sa version.)

## Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération

Problème : lorsque vous exécutez une génération, le journal de génération contient cet avertissement.

Cause possible : votre build n'utilise pas la version 1.0 ou ultérieure de l'image standard Amazon Linux 2 (AL2), ni la version 2.0 ou ultérieure de l'image standard Ubuntu, et un environnement d'exécution est spécifié dans une `runtime-versions` section de votre fichier buildspec.

Solution recommandée : vérifiez que votre fichier buildspec ne contient pas une section `runtime-versions`. La `runtime-versions` section n'est obligatoire que si vous utilisez l'image standard Amazon Linux 2 (AL2) ou version ultérieure ou l'image standard Ubuntu version 2.0 ou ultérieure.

## Erreur : « Impossible de vérifier l' JobWorker identité » lors de l'ouverture de la CodeBuild console

Problème : Lorsque vous ouvrez la CodeBuild console, le message d'erreur « Impossible de vérifier l' JobWorker identité » s'affiche.

Cause possible : le rôle IAM utilisé pour accéder à la console possède une balise utilisée `jobId` comme clé. Cette clé de balise est réservée CodeBuild et provoquera cette erreur si elle est présente.

Solution recommandée : modifiez toutes les balises de rôle IAM personnalisées contenant la clé `jobId` pour qu'elles aient une clé différente, par exemple `jobIdentifier`.

## La construction n'a pas pu démarrer

Problème : Lorsque vous démarrez une compilation, vous recevez un message d'erreur indiquant que la compilation n'a pas pu démarrer.

Cause possible : le nombre de versions simultanées a été atteint.

Solutions recommandées : attendez que les autres versions soient terminées ou augmentez la limite de génération simultanée pour le projet, puis recommencez la génération. Pour de plus amples informations, veuillez consulter [Configuration du projet](#).

## Accès aux GitHub métadonnées dans les versions mises en cache localement

Problème : Dans certains cas, le répertoire `.git` d'une version mise en cache est un fichier texte et non un répertoire.

Causes possibles : lorsque la mise en cache des sources locales est activée pour une compilation, CodeBuild crée un gitlink pour le `.git` répertoire. Cela signifie que le `.git` répertoire est en fait un fichier texte contenant le chemin d'accès au répertoire.

Solutions recommandées : Dans tous les cas, utilisez la commande suivante pour obtenir le répertoire de métadonnées Git. Cette commande fonctionnera quel que soit le format de `.git` :

```
git rev-parse --git-dir
```

## AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3...

Problème : lors du téléchargement de données de test dans un compartiment Amazon S3, CodeBuild il est impossible d'écrire les données de test dans le compartiment.

### Causes possibles :

- Le compte spécifié pour le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment Amazon S3.
- Le rôle de service ne dispose pas d'un accès en écriture au compartiment.

### Solutions recommandées :

- Modifiez le propriétaire du compartiment du groupe de rapports pour qu'il corresponde au propriétaire du compartiment Amazon S3.
- Modifiez le rôle de service pour autoriser l'accès en écriture au compartiment Amazon S3.

## Erreur : « Vos informations d'identification ne disposent pas d'une ou de plusieurs étendues de privilèges requises » lors de la création d'un CodeBuild projet avec CodeConnections

Problème : lorsque vous créez un CodeBuild projet avec CodeConnections, vous n'êtes pas autorisé à installer un webhook Bitbucket.

### Causes possibles :

- La nouvelle étendue d'autorisation n'a peut-être pas été acceptée dans votre compte Bitbucket.

### Solutions recommandées :

- Pour accepter la nouvelle autorisation, vous devez avoir reçu un e-mail avec un objet intitulé Action requise - Scopes for AWS CodeStar have changed envoyé par Bitbucket, notifications-noreply@bitbucket.org L'e-mail contient un lien pour autoriser le webhook à accéder à votre installation d'application CodeConnections Bitbucket existante.
- Si vous ne trouvez pas l'e-mail, vous pouvez accorder l'autorisation en accédant ou [https://bitbucket.org/site/addons/reauthorize?addon\\_key=aws-codestar](https://bitbucket.org/site/addons/reauthorize?addon_key=aws-codestar) en sélectionnant l'espace de travail auquel vous souhaitez autoriser le webhook. [https://bitbucket.org/site/addons/reauthorize?account=<workspace-name>&addon\\_key=aws-codestar](https://bitbucket.org/site/addons/reauthorize?account=<workspace-name>&addon_key=aws-codestar)

**AWS CodeStar requests access**

This app is hosted at <https://codestar-connections.webhooks.aws>

Read your account information

Read and modify your repositories and their pull requests

Administer your repositories

Read and modify your repositories' webhooks

Authorize for workspace

**Allow AWS CodeStar to do this?**

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

**Grant access** Cancel

Erreur : « Désolé, aucun terminal n'a été demandé, impossible d'obtenir des informations » lors de la compilation avec la commande d'installation d'Ubuntu

Problème : si vous utilisez des versions privilégiées du conteneur GPU, vous êtes peut-être en train d'installer le NVIDIA Container Toolkit en suivant cette [procédure](#). Dans la dernière version CodeBuild d'image, CodeBuild préinstalle et configure docker nvidia-container-toolkit avec l'image la amazonlinux plus récente et organisée. ubuntu Si vous suivez cette procédure, les builds utilisant la commande d'installation d'Ubuntu échoueront avec l'erreur suivante :

```
Running command curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | gpg --
dearmor --no-tty -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
gpg: Sorry, no terminal at all requested - can't get input
curl: (23) Failed writing body
```

Causes possibles : La clé GPG existe déjà au même endroit.



Solutions recommandées : `nvidia-container-toolkit` Il est déjà installé dans l'image. Si cette erreur s'affiche, vous pouvez ignorer le processus d'installation et de redémarrage de docker dans votre buildspec.

# Quotas pour AWS CodeBuild

Les tableaux suivants répertorient les quotas actuels dans AWS CodeBuild. Ces quotas s'appliquent à chaque AWS région prise en charge et à chaque AWS compte, sauf indication contraire.

## Quotas de service

Les quotas par défaut pour le AWS CodeBuild service sont les suivants.

| Name                                                                | Par défaut                            | Ajusté              | Description                                                                                                                            |
|---------------------------------------------------------------------|---------------------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Tags associés par projet                                            | Chaque région prise en charge : 50    | Non                 | Nombre maximum de balises que vous pouvez associer à un projet de génération                                                           |
| Construisez des projets                                             | Chaque région prise en charge : 5 000 | <a href="#">Oui</a> | Nombre maximal de projets de version                                                                                                   |
| Délai d'expiration du délai d'attente en minutes                    | Chaque région prise en charge : 2 160 | Non                 | Délai de construction maximal en minutes                                                                                               |
| Demande simultanée d'informations sur les builds                    | Chaque région prise en charge : 100   | Non                 | Nombre maximum de versions pour lesquelles vous pouvez demander des informations à tout moment à l'aide de la AWS CLI ou d'un AWS SDK. |
| Demandes simultanées d'informations sur les projets de construction | Chaque région prise en charge : 100   | Non                 | Nombre maximum de projets de construction pour lesquels vous pouvez demander des informations à tout                                   |

| Name                                                                   | Par défaut                     | Ajuste              | Description                                                                              |
|------------------------------------------------------------------------|--------------------------------|---------------------|------------------------------------------------------------------------------------------|
|                                                                        |                                |                     | moment à l'aide de la AWS CLI ou d'un AWS SDK.                                           |
| Exécution simultanée de versions pour l'environnement ARM Lambda/10 Go | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/10 Go |
| Exécution simultanée de versions pour l'environnement ARM Lambda/1 Go  | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/1 Go  |
| Exécution simultanée de versions pour l'environnement ARM Lambda/2 Go  | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/2 Go  |
| Exécution simultanée de versions pour l'environnement ARM Lambda/4 Go  | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/4 Go  |
| Exécution simultanée de versions pour l'environnement ARM Lambda/8 Go  | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/8 Go  |

| Name                                                                   | Par défaut                        | Ajustable           | Description                                                                             |
|------------------------------------------------------------------------|-----------------------------------|---------------------|-----------------------------------------------------------------------------------------|
| Exécution simultanée de versions pour l'environnement ARM/2 XLarge     | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM/2 XLarge    |
| Exécution simultanée de versions pour l'environnement ARM/Large        | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM/Large       |
| Exécution simultanée de versions pour l'environnement ARM/Medium       | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM/Medium      |
| Exécution simultanée de versions pour l'environnement ARM/Small        | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM/Small       |
| Exécution simultanée de versions pour l'environnement ARM/XLarge       | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement ARM/XLarge      |
| Exécution simultanée de versions pour un environnement Linux GPU Large | Chaque région prise en charge : 0 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux GPU/Large |

| Name                                                                     | Par défaut                        | Ajuste              | Description                                                                                |
|--------------------------------------------------------------------------|-----------------------------------|---------------------|--------------------------------------------------------------------------------------------|
| Exécution simultanée de versions pour un environnement Linux GPU Small   | Chaque région prise en charge : 0 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux GPU/Small    |
| Exécution simultanée de versions pour l'environnement Linux Lambda/10 Go | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/10 Go |
| Exécution simultanée de versions pour l'environnement Linux Lambda/1 Go  | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/1 Go  |
| Exécution simultanée de versions pour l'environnement Linux Lambda/2 Go  | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/2 Go  |
| Exécution simultanée de versions pour l'environnement Linux Lambda/4 Go  | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/4 Go  |
| Exécution simultanée de versions pour l'environnement Linux Lambda/8 Go  | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/8 Go  |

| Name                                                                             | Par défaut                        | Ajuste              | Description                                                                                       |
|----------------------------------------------------------------------------------|-----------------------------------|---------------------|---------------------------------------------------------------------------------------------------|
| Exécution simultanée de versions pour l'environnement Linux/2 XLarge             | Chaque région prise en charge : 0 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/2 XLarge            |
| Exécution simultanée de versions pour l'environnement Linux/Large                | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Large               |
| Exécution simultanée de versions pour l'environnement Linux/Medium               | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Medium              |
| Exécution simultanée de versions pour l'environnement Linux/Small                | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Small               |
| Exécution simultanée de versions pour l'environnement Linux/XLarge               | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/XLarge              |
| Exécution simultanée de versions pour l'environnement Windows Server 2019/ Large | Par région prise en charge : 1    | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2019/Large |

| Name                                                                               | Par défaut                     | Ajusté              | Description                                                                                          |
|------------------------------------------------------------------------------------|--------------------------------|---------------------|------------------------------------------------------------------------------------------------------|
| Exécution simultanée de versions pour l'environnement Windows Server 2019/Medium   | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2019/Medium   |
| Exécution simultanée de versions pour l'environnement Windows Server 2022/2 XLarge | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2022/2 XLarge |
| Exécution simultanée de versions pour un environnement Windows Server 2022/Large   | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour un environnement Windows Server 2022/Large   |
| Exécution simultanée de versions pour l'environnement Windows Server 2022/Medium   | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2022/Medium   |
| Exécution simultanée de versions pour l'environnement Windows Server 2022/XLarge   | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2022/ XLarge  |
| Exécution simultanée de versions pour l'environnement Windows/Large                | Par région prise en charge : 1 | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows/Large                |

| Name                                                                 | Par défaut                         | Ajuste              | Description                                                                            |
|----------------------------------------------------------------------|------------------------------------|---------------------|----------------------------------------------------------------------------------------|
| Exécution simultanée de versions pour l'environnement Windows/Medium | Par région prise en charge : 1     | <a href="#">Oui</a> | Nombre maximum de versions exécutées simultanément pour l'environnement Windows/Medium |
| Période minimale d'expiration de la construction en minutes          | Chaque région prise en charge : 5  | Non                 | Délai de construction minimal en minutes                                               |
| Groupe de sécurité dans le cadre d'une configuration VPC             | Chaque région prise en charge : 5  | Non                 | Groupe de sécurité disponibles pour la configuration VPC                               |
| Sous-réseaux sous configuration VPC                                  | Chaque région prise en charge : 16 | Non                 | Sous-réseaux disponibles pour la configuration VPC                                     |

 Note

Les métriques internes détermineront les quotas par défaut pour les builds exécutés simultanément.

Les quotas pour le nombre maximum de builds exécutés simultanément varient en fonction du type de calcul. Pour certaines plateformes et certains types de calcul, la valeur par défaut est 20. Pour demander un quota de construction simultanée plus élevé, ou si le message d'erreur « Impossible d'avoir plus de X versions actives pour le compte » s'affiche, utilisez le lien ci-dessus pour effectuer la demande. Pour plus d'informations sur les tarifs, consultez la section [AWS CodeBuild tarification](#).



## Autres limites

### Construisez des projets

| Ressource                                                                                                                                             | Par défaut                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Caractères autorisés dans une description de projet de génération                                                                                     | N'importe lequel                                                            |
| Caractères autorisés dans un nom de projet de génération                                                                                              | Les lettres A-Z et a-z, les chiffres 0-9, et les caractères spéciaux - et _ |
| Longueur d'un nom de projet de génération                                                                                                             | 2 à 150 caractères inclus                                                   |
| Longueur maximum d'une description de projet de génération                                                                                            | 255 caractères                                                              |
| Nombre maximal de rapports que vous pouvez ajouter à un projet                                                                                        | 5                                                                           |
| Nombre de minutes que vous pouvez spécifier comme délai d'expiration de génération dans un projet de génération pour toutes les générations associées | 5 à 21 h 60 (36 heures)                                                     |

### Construit

| Ressource                                                                                                     | Par défaut              |
|---------------------------------------------------------------------------------------------------------------|-------------------------|
| Durée maximale de conservation de l'historique d'une génération                                               | 1 an                    |
| Nombre de minutes que vous pouvez spécifier comme délai d'expiration de génération dans une génération unique | 5 à 21 h 60 (36 heures) |

## Flottes informatiques

| Ressource                                                                                       | Par défaut |
|-------------------------------------------------------------------------------------------------|------------|
| Nombre simultané de flottes informatiques                                                       | 10         |
| Exécution simultanée d'instances pour des flottes d' ARM/Small environnements                   | 1          |
| Exécution simultanée d'instances pour des flottes d' ARM/Large environnements                   | 1          |
| Exécution simultanée d'instances pour des flottes d' Linux/Small environnements                 | 1          |
| Exécution simultanée d'instances pour des flottes d' Linux/Medium environnements                | 1          |
| Exécution simultanée d'instances pour des flottes d' Linux/Large environnements                 | 1          |
| Exécution simultanée d'instances pour des flottes d' Linux/XLarge environnements                | 1          |
| Exécution simultanée d'instances pour les flottes d'environnements Linux/2 XLarge               | 0          |
| Instances exécutées simultanément pour les flottes d' GPU/Small environnements Linux            | 0          |
| Instances exécutées simultanément pour les flottes d' GPU/Large environnements Linux            | 0          |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2019/ Medium | 1          |

| Ressource                                                                                       | Par défaut |
|-------------------------------------------------------------------------------------------------|------------|
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2019/ Large  | 1          |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2022/ Medium | 1          |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2022/ Large  | 1          |
| Exécution simultanée d'instances pour les flottes d' ARM/Medium environnements Mac              | 1          |
| Exécution simultanée d'instances pour les flottes d' ARM/Large environnements Mac               | 1          |

## Rapports

| Ressource                                                            | Par défaut       |
|----------------------------------------------------------------------|------------------|
| Durée maximale d'un rapport de test est disponible après sa création | 30 jours         |
| Longueur maximale d'un message de scénario de test                   | 5 000 caractères |
| Longueur maximale du nom d'un scénario de test                       | 1 000 caractères |
| Nombre maximum de groupes de rapports par AWS compte                 | 5 000            |
| Nombre maximal de cas de test par rapport au rapport                 | 500              |

## Étiquettes

Les limites de balises s'appliquent aux balises figurant sur les projets de CodeBuild création et les ressources des groupes de CodeBuild rapports.

| Ressource                           | Par défaut                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Noms de clés de balise de ressource | <p>Toute combinaison de lettres, chiffres, espaces au format Unicode et caractères UTF-8 autorisés avec une longueur de 1 à 127 caractères. Les caractères autorisés sont les suivants : + - = . _ : / @</p> <p>Les noms de clé de balise doivent être uniques, et chaque clé ne peut avoir qu'une seule valeur. Un nom de clé de balise ne peut pas :</p> <ul style="list-style-type: none"> <li>commencer par aws :</li> <li>être composée uniquement d'espaces</li> <li>se terminer par un espace</li> <li>contenir des émoticônes ou l'un des caractères suivants : ? ^ * [ \ ~ ! # \$ % &amp; * ( ) &gt; &lt;   " ' ` [ ] { } ;</li> </ul> |
| Valeurs de la balise de ressource.  | <p>Toute combinaison de lettres, chiffres, espaces au format Unicode et caractères UTF-8 autorisés avec une longueur de 0 à 255 caractères. Les caractères autorisés sont les suivants : + - = . _ : / @</p> <p>Une clé ne peut avoir qu'une seule valeur, mais plusieurs clés peuvent avoir la même valeur. Une valeur de clé de balise ne peut pas contenir d'émoticônes ou l'un des caractères suivants : ? ^ * [ \ ~ ! # \$ % &amp; * ( ) &gt; &lt;   " ' ` [ ] { } ;</p>                                                                                                                                                                   |

# AWS CodeBuild Historique du document du guide de l'utilisateur

Le tableau suivant décrit les modifications importantes apportées à la documentation depuis la dernière version de AWS CodeBuild. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

- Dernière version de l'API : 2016-10-06

| Modification                                                                                | Description                                                                                                                                                                                                              | Date              |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Contenu mis à jour : politiques AWS gérées (prédéfinies) pour AWS CodeBuild</a> | Les AWSCode BuildDeveloperAccess politiques AWSCode BuildAdminAccess et ont été mises à jour pour inclure l'ssmessages:OpenDataChannel autorisation requise pour le débogage interactif des versions de Session Manager. | 1er décembre 2025 |
| <a href="#">Approbation des commentaires par pull request</a>                               | CodeBuild prend en charge les politiques de génération de pull request qui fournissent un contrôle supplémentaire sur les builds déclenchés par des pull requests.                                                       | 7 août 2025       |
| <a href="#">Contenu mis à jour : politiques AWS gérées (prédéfinies) pour AWS CodeBuild</a> | Les AWSCode BuildReadOnlyAccess politiques AWSCodeBuildAdminAccess, AWSCodeBuildDeveloperAccess, et ont été mises à jour. La ressource d'origine arn:aws:codeconnections:*:*:connecti                                    | 10 juin 2025      |

on/\* a été mise à jour  
versarn:aws:codeconnec  
tions:\*:\*:\* .

### [Nouvelle référence pour les clés de CodeBuild condition](#)

Ajout d'une nouvelle page de référence contenant des exemples d'utilisation des clés de CodeBuild condition. Voir les [clés de AWS CodeBuild condition](#).

15 mai 2025

### [Nouveau contenu : Exemple de serveur de génération d'images Docker pour CodeBuild](#)

CodeBuild prend désormais en charge le téléchargement de vos versions Docker vers un serveur de génération d'images géré.

15 mai 2025

### [Nouveau type de calcul : CUSTOM\\_INSTANCE\\_TYPE](#)

CodeBuild vous permet désormais de créer un parc de capacités réservées avec un type d'instance spécifique en utilisantCUSTOM\_INSTANCE\_TYPE .

23 avril 2025

### [Nouveau support pour CodeBuild sandbox](#)

Ajout d'informations sur l'utilisation du nouveau CodeBuild bac à sable. Voir [Déboguer les versions avec CodeBuild sandbox](#).

7 avril 2025

### [Nouveaux types d'environnement Windows](#)

CodeBuild prend désormais en charge les types d'environnement Windows XL et 2XL. Pour de plus amples informations, veuillez consulter [Types de calcul de l'environnement de génération](#).

31 mars 2025

|                                                                                                                              |                                                                                                                                                           |                 |
|------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Mise à jour de la mise en cache d'Amazon S3</a>                                                                  | CodeBuild prend désormais en charge un nouveau comportement de mise en cache pour la mise en cache d'Amazon S3.                                           | 28 mars 2025    |
| <a href="#">Nouveau contenu : options de configuration d' GitHub Actions Runner</a>                                          | CodeBuild prend désormais en <code>CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME</code> charge l'enregistrement au niveau de l'entreprise. | 11 mars 2025    |
| <a href="#">Nouveau contenu : ajout d'un nouveau type de filtre Webhook</a>                                                  | Ajoutez la prise en charge d'un nouveau type de filtre Webhook ( <code>ORGANIZATION_NAME</code> ).                                                        | 11 mars 2025    |
| <a href="#">Nouveau contenu : Tutoriel pour la signature de code Apple avec Fastlane avec stockage de certificats S3</a>     | Ajout d'un nouveau didacticiel pour la signature de code Apple avec Fastlane lors de CodeBuild l'utilisation de S3 pour le stockage de certificats        | 5 février 2025  |
| <a href="#">Nouveau contenu : Tutoriel pour la signature de code Apple avec Fastlane avec GitHub stockage de certificats</a> | Ajout d'un nouveau didacticiel pour la signature de code Apple avec Fastlane lors de CodeBuild l'utilisation GitHub pour le stockage de certificats       | 5 février 2025  |
| <a href="#">Nouveau contenu : Buildkite Runner</a>                                                                           | Ajouter du nouveau contenu pour le Buildkite Runner                                                                                                       | 31 janvier 2025 |
| <a href="#">Nouveau contenu : webhooks manuels Buildkite</a>                                                                 | Ajoutez le support pour les webhooks manuels Buildkite.                                                                                                   | 31 janvier 2025 |

|                                                                                                               |                                                                                                                                                                                                                 |                  |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Nouveau contenu : référence Batch build buildspec</a>                                             | Ajoutez la prise en charge des builds par lots dans les flottes de capacités réservées et les environnements Lambda.                                                                                            | 8 janvier 2025   |
| <a href="#">Nouveau contenu : Exécuter des tests parallèles dans les builds par lots</a>                      | Ajoutez du nouveau contenu pour les tests parallèles dans les versions par lots.                                                                                                                                | 2 janvier 2025   |
| <a href="#">Nouveau contenu : Retry construit automatiquement</a>                                             | CodeBuild prend désormais en charge la nouvelle tentative automatique pour les versions de webhook.                                                                                                             | 18 décembre 2024 |
| <a href="#">Nouveau contenu : configurer un identifiant de registre privé pour les coureurs auto-hébergés</a> | Ajoutez la prise en charge de la définition des informations d'identification du registre lors de l'utilisation d'images personnalisées provenant de registres non privés.                                      | 13 décembre 2024 |
| <a href="#">Nouveau contenu : options de configuration d' GitHub Actions Runner</a>                           | CodeBuild GitHub Les actions auto-hébergées par les coureurs vous permettent désormais d'enregistrer vos coureurs au niveau de l'organisation et de configurer un identifiant de groupe de coureurs spécifique. | 12 décembre 2024 |
| <a href="#">Nouveau contenu : ajout d'un attribut en cas de défaillance RETRY</a>                             | CodeBuild vous permet désormais de configurer un attribut en cas d'échec RETRY dans votre buildspec.                                                                                                            | 12 décembre 2024 |
| <a href="#">Nouveau contenu : GitLab webhooks manuels</a>                                                     | Ajoutez le support pour les webhooks GitLab manuels.                                                                                                                                                            | 11 décembre 2024 |



|                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                 |                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Contenu mis à jour : alias mis à jour</a>                                                                                             | Mettez à jour les alias pour les images d'exécution standard basées sur Linux.                                                                                                                                                                                                                                                  | 22 novembre 2024 |
| <a href="#">Contenu mis à jour : les remplacements d'étiquettes sont pris en charge par le CodeBuild coureur hébergé GitLab</a>                   | Ajoutez la prise en charge des remplacements d'étiquettes illustrées personnalisées pour les GitLab coureurs.                                                                                                                                                                                                                   | 22 novembre 2024 |
| <a href="#">Contenu mis à jour : les remplacements d'étiquettes sont pris en charge avec le programme Actions CodeBuild Runner hébergé GitHub</a> | Ajoutez la prise en charge des remplacements d'étiquettes d'image personnalisés pour les coureurs GitHub Actions.                                                                                                                                                                                                               | 22 novembre 2024 |
| <a href="#">Contenu mis à jour : politiques AWS gérées (prédéfinies) pour AWS CodeBuild</a>                                                       | Les AWSCode BuildRead OnlyAccess politiques AWSCodeBuildAdminAccess, AWSCodeBuildDeveloperAccess, et ont été mises à jour. La ressource d'origine <code>arn:aws:codebuild:</code> <ul style="list-style-type: none"><li>* a été mise à jour vers <code>arn:aws:codebuild:</code></li><li>* <code>::project/*</code> .</li></ul> | 15 novembre 2024 |
| <a href="#">Contenu mis à jour : capacité réservée</a>                                                                                            | Les flottes de capacité réservée prennent désormais en charge les versions autres que des conteneurs : ARM EC2, Linux EC2 et Windows EC2.                                                                                                                                                                                       | 12 novembre 2024 |

|                                                                                                                               |                                                                                                 |                   |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------------------|
| <a href="#"><u>Contenu mis à jour : capacité réservée</u></a>                                                                 | Les flottes de capacité réservée prennent désormais en charge le calcul basé sur les attributs. | 6 novembre 2024   |
| <a href="#"><u>Nouveau contenu : Retry construit automatiquement</u></a>                                                      | CodeBuild vous permet désormais d'activer la nouvelle tentative automatique pour vos builds.    | 25 octobre 2024   |
| <a href="#"><u>Nouveau contenu : Exécuter CodeBuild sur un serveur proxy géré pour les flottes de capacité réservée</u></a>   | Ajoutez la prise en charge des configurations proxy pour les flottes de capacité réservée.      | 15 octobre 2024   |
| <a href="#"><u>Nouveau contenu : coureurs autogérés GitLab</u></a>                                                            | Ajoutez du nouveau contenu pour les coureurs autogérés GitLab                                   | 17 septembre 2024 |
| <a href="#"><u>Nouveau contenu : GitLab webhooks de groupe</u></a>                                                            | Ajoutez la prise en charge des webhooks de GitLab groupe.                                       | 17 septembre 2024 |
| <a href="#"><u>Nouveau contenu : exécutez les commandes buildspec pendant les phases INSTALL, PRE_BUILD et POST_BUILD</u></a> | Ajoutez le support pour <code>with-buildspec</code> .                                           | 20 août 2024      |
| <a href="#"><u>Contenu mis à jour : capacité réservée</u></a>                                                                 | Les flottes de capacité réservée sont désormais compatibles avec macOS.                         | 19 août 2024      |
| <a href="#"><u>Nouveau contenu : connexions aux GitHub applications</u></a>                                                   | Ajoutez la prise en charge des connexions aux GitHub applications.                              | 14 août 2024      |
| <a href="#"><u>Nouveau contenu : connexions à l'application Bitbucket</u></a>                                                 | Ajoutez la prise en charge des connexions à l'application Bitbucket.                            | 14 août 2024      |

|                                                                                |                                                                                                                                                                                                                         |                 |
|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Nouveau contenu : Plusieurs jetons d'accès dans CodeBuild</a>      | Ajoutez la prise en charge de l'approvisionnement en jetons d'accès auprès de fournisseurs tiers à partir de secrets contenus dans AWS Secrets Manager ou via AWS CodeConnections des connexions.                       | 14 août 2024    |
| <a href="#">Contenu mis à jour : capacité réservée</a>                         | Les flottes de capacité réservée prennent désormais en charge les types de XLarge calcul ARM Medium XLarge, ARM et ARM 2.                                                                                               | 5 août 2024     |
| <a href="#">Contenu mis à jour : capacité réservée</a>                         | CodeBuild prend désormais en charge la connectivité VPC pour les flottes de capacité réservée sous Windows.                                                                                                             | 1er août 2024   |
| <a href="#">Nouveaux types de calcul ARM</a>                                   | CodeBuild prend désormais en charge les types de XLarge calcul ARM Medium XLarge, ARM et ARM 2. Pour de plus amples informations, veuillez consulter <a href="#">Types de calcul de l'environnement de génération</a> . | 10 juillet 2024 |
| <a href="#">Contenu mis à jour : signature SHA</a>                             | Mettez à jour la signature de l'algorithme de hachage sécurisé (SHA) pour le x86_64 et l'ARM.                                                                                                                           | 19 juin 2024    |
| <a href="#">Nouveau contenu : GitHub webhooks mondiaux et organisationnels</a> | Ajoutez la prise en charge des webhooks GitHub mondiaux et organisationnels.                                                                                                                                            | 17 juin 2024    |

|                                                                                             |                                                                                                                                                                           |               |
|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">Nouveau contenu : ajout d'un nouveau type de filtre Webhook</a>                 | Ajoutez la prise en charge d'un nouveau type de filtre Webhook (REPOSITORY_NAME ).                                                                                        | 17 juin 2024  |
| <a href="#">Espace disque mis à jour</a>                                                    | Les types ARM Small de ARM Large calcul et de calcul disposent désormais d'un espace disque accru.                                                                        | 4 juin 2024   |
| <a href="#">Nouveau contenu : GitHub webhooks manuels</a>                                   | Ajoutez le support pour les webhooks GitHub manuels.                                                                                                                      | 23 mai 2024   |
| <a href="#">Contenu mis à jour : capacité réservée</a>                                      | CodeBuild prend désormais en charge la connectivité VPC pour les flottes de capacité réservée sur Amazon Linux.                                                           | 15 mai 2024   |
| <a href="#">Contenu mis à jour : Images de calcul Lambda</a>                                | Ajouter le support Lambda pour .NET 8 (a1-lambda/aarch64/dotnet8 et a1-lambda/x86_64/dotnet8                                                                              | 8 mai 2024    |
| <a href="#">Quota mis à jour : délai de construction</a>                                    | Mettez à jour le quota de délai d'expiration maximal de construction à 2 160 minutes (36 heures).                                                                         | 1 mai 2024    |
| <a href="#">Contenu mis à jour : politiques AWS gérées (prédéfinies) pour AWS CodeBuild</a> | Les AWSCode BuildReadOnlyAccess, AWSCodeBuildAdminAccess, AWSCodeBuildDeveloperAccess, et ont été mises à jour pour refléter le changement de AWS CodeConnections marque. | 30 avril 2024 |

|                                                                                                           |                                                                                                                                                 |                 |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Nouveau contenu : Mot de passe ou jeton d'accès de l'application Bitbucket</a>                | Ajoutez la prise en charge des jetons d'accès Bitbucket.                                                                                        | 11 avril 2024   |
| <a href="#">Nouveau contenu : Découvrez automatiquement les rapports dans CodeBuild</a>                   | CodeBuild prend désormais en charge la découverte automatique des rapports.                                                                     | 4 avril 2024    |
| <a href="#">Nouveau contenu : GitHub Actions runners auto-hébergés</a>                                    | Ajoutez du nouveau contenu pour les coureurs GitHub Actions auto-hébergés                                                                       | 2 avril 2024    |
| <a href="#">Nouveau contenu : GitLab connexions</a>                                                       | Ajoutez la prise en charge GitLab et les connexions GitHub autogérées.                                                                          | 25 mars 2024    |
| <a href="#">Nouveau contenu : ajout de nouveaux événements de webhook et de nouveaux types de filtres</a> | Ajoutez la prise en charge des nouveaux événements webhook (RELEASEDetPRERELEASED ) et des nouveaux types de filtres (TAG_NAMEetRELEASE_NAME ). | 15 mars 2024    |
| <a href="#">Nouveau contenu : Ajoutez un nouvel événement webhook : PULL_REQUEST_CLOSED</a>               | Ajouter le support pour un nouvel événement webhook :PULL_REQUEST_CLOSED .                                                                      | 20 février 2024 |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a>                                 | Ajouter le support pour Windows Server Core 2019 (windows-base:2019-3.0 )                                                                       | 7 février 2024  |

|                                                                           |                                                                                                                                                                                      |                  |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a> | Ajout de la prise en charge des nouveaux environnements d'exécution pour Amazon Linux 2023 ( ) a12/aarch64/standard/3.0                                                              | 29 janvier 2024  |
| <a href="#">Nouveau contenu : capacité réservée</a>                       | CodeBuild prend désormais en charge les flottes de capacité réservée en. CodeBuild                                                                                                   | 18 janvier 2024  |
| <a href="#">Nouveau type de calcul</a>                                    | CodeBuild supporte désormais un type de XLarge calcul Linux. Pour de plus amples informations, veuillez consulter <a href="#">Types de calcul de l'environnement de génération</a> . | 8 janvier 2024   |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a> | Ajout de la prise en charge des nouveaux environnements d'exécution pour Amazon Linux (2a12/standard/5.0 ) et Ubuntu (ubuntu/standard/7.0 )                                          | 14 décembre 2023 |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a> | Ajout de la prise en charge des nouvelles images de calcul Lambda                                                                                                                    | 8 décembre 2023  |
| <a href="#">Nouveau contenu : AWS Lambda calcul</a>                       | Ajouter du nouveau contenu pour le AWS Lambda calcul                                                                                                                                 | 6 novembre 2023  |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a> | Ajouter le support pour Amazon Linux 2 (a12/standard/5.0 )                                                                                                                           | 17 mai 2023      |

|                                                                              |                                                                                                                                                                                                                                         |               |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">Modifications apportées aux politiques gérées pour CodeBuild</a> | Les informations relatives aux mises à jour des politiques AWS gérées pour CodeBuild sont désormais disponibles. Pour plus d'informations, consultez la rubrique <a href="#">Mises à jour CodeBuild des politiques gérées par AWS</a> . | 16 mai 2023   |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a>    | Supprimer le support pour Amazon Linux 2 (a12/standard/3.0 ) et ajouter le support pour Amazon Linux 2 (a12/standard/corretto8 ) et Amazon Linux 2 (a12/standard/corretto11 )                                                           | 9 mai 2023    |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a>    | Ajouter le support pour Ubuntu 22.04 ( ) ubuntu/standard/7.0                                                                                                                                                                            | 13 avril 2023 |
| <a href="#">Contenu mis à jour : Images Docker fournies par CodeBuild</a>    | Supprimer le support pour Ubuntu 18.04 (ubuntu/standard/4.0 ) et Amazon Linux 2 ( ) a12/aarch64/standard/1.0                                                                                                                            | 31 mars 2023  |

[Contenu mis à jour : suppression de la limitation VPC](#)

Suppression de la limitation suivante : si vous configurez CodeBuild pour fonctionner avec un VPC, la mise en cache locale n'est pas prise en charge. À compter du 28/02/22, le build de votre VPC prendra plus de temps car une nouvelle instance Amazon EC2 sera utilisée pour chaque build.

1er mars 2023

[Contenu mis à jour : Images Docker fournies par CodeBuild](#)

Supprimer le support pour Ubuntu 18.04 (ubuntu/standard/3.0 ) et Amazon Linux 2 () al2/standard/2.0

30 juin 2022

[Amazon ECR Sample : Restreindre l'accès aux images](#)

Lorsque les CodeBuild informations d'identification sont utilisées pour extraire une image Amazon ECR, vous pouvez restreindre l'accès à l'image à un CodeBuild projet spécifique. Pour plus d'informations, consultez l'[exemple Amazon ECR](#).

10 mars 2022



### [Prise en charge de régions supplémentaires](#)

Le type de ARM\_CONTAINER calcul est désormais pris en charge dans les régions supplémentaires suivantes : Asie-Pacifique (Séoul), Canada (centre), Europe (Londres) et Europe (Paris). Pour de plus amples informations, veuillez consulter [Types de calcul de l'environnement de génération](#).

10 mars 2022

### [Nouvelle limitation en matière de VPC](#)

Si vous configurez CodeBuild pour fonctionner avec un VPC, la mise en cache locale n'est pas prise en charge. À compter du 28/02/22, le build de votre VPC prendra plus de temps car une nouvelle instance Amazon EC2 sera utilisée pour chaque build.

25 février 2022

### [Mode de rapport par lots](#)

CodeBuild vous permet désormais de sélectionner la manière dont les statuts de génération par lots sont envoyés au fournisseur source d'un projet. Pour plus d'informations, consultez la section [Mode de rapport Batch](#).

4 octobre 2021

|                                                                       |                                                                                                                                                                                                                                                                                 |                   |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Nouveau type de calcul</a>                                | CodeBuild prend désormais en charge un petit type de calcul ARM. Pour de plus amples informations, veuillez consulter <a href="#">Types de calcul de l'environnement de génération</a> .                                                                                        | 13 septembre 2021 |
| <a href="#">Projets de construction publics</a>                       | CodeBuild vous permet désormais de mettre les résultats de construction de vos projets de construction à la disposition du public sans avoir besoin d'accéder à un AWS compte. Pour plus d'informations, consultez la section <a href="#">Projets de construction publics</a> . | 11 août 2021      |
| <a href="#">Débogage de session pour les compilations par lots</a>    | CodeBuild prend désormais en charge le débogage de session pour les compilations par lots. <a href="#">Pour plus d'informations, consultez build-graph et build-list</a> .                                                                                                      | 3 mars 2021       |
| <a href="#">Limite de construction simultanée au niveau du projet</a> | CodeBuild vous permet désormais de limiter le nombre de builds simultanés pour un projet de build. Pour plus d'informations, voir <a href="#">Configuration du projet</a> et <a href="#">concurrentBuildLimit</a> .                                                             | 16 février 2021   |

|                                                                |                                                                                                                                                                                                                                                                                      |                |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">Nouvelle propriété buildspec : s3-prefix</a>       | CodeBuild fournit désormais la propriété <code>s3-prefix</code> buildspec pour les artefacts qui vous permet de spécifier un préfixe de chemin pour les artefacts chargés sur Amazon S3. Pour plus d'informations, consultez la section <a href="#">s3-prefix</a> .                  | 9 février 2021 |
| <a href="#">Nouvelle propriété buildspec : en cas d'échec</a>  | CodeBuild fournit désormais la propriété <code>on-failure</code> buildspec pour les phases de construction qui vous permet de déterminer ce qui se passe en cas d'échec d'une phase de construction. Pour plus d'informations, consultez la section <a href="#">en cas d'échec</a> . | 9 février 2021 |
| <a href="#">Nouvelle propriété buildspec : exclude-paths</a>   | CodeBuild fournit désormais la propriété <code>exclude-paths</code> buildspec pour les artefacts qui vous permet d'exclure des chemins de vos artefacts de construction. Pour plus d'informations, consultez <a href="#">exclude-paths</a> .                                         | 9 février 2021 |
| <a href="#">Nouvelle propriété buildspec : enable-symlinks</a> | CodeBuild fournit désormais la propriété <code>enable-symlinks</code> buildspec pour les artefacts qui vous permet de conserver les liens symboliques dans un artefact ZIP. Pour plus d'informations, consultez la section <a href="#">enable-symlinks</a> .                         | 9 février 2021 |

|                                                             |                                                                                                                                                                                                |                 |
|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Amélioration du nom de l'artefact Buildspec</a> | CodeBuild permet désormais à la <code>artifacts/name</code> propriété de contenir des informations de chemin. Pour plus d'informations, voir <a href="#">le nom</a> .                          | 9 février 2021  |
| <a href="#">Rapports sur la couverture du code</a>          | CodeBuild fournit désormais des rapports sur la couverture du code. Pour plus d'informations, consultez les <a href="#">rapports de couverture du code</a> .                                   | 30 juillet 2020 |
| <a href="#">Constructions par lots</a>                      | CodeBuild prend désormais en charge l'exécution de versions simultanées et coordonnées d'un projet. Pour plus d'informations, consultez la section <a href="#">Batch builds in CodeBuild</a> . | 30 juillet 2020 |
| <a href="#">Image Windows Server 2019</a>                   | CodeBuild fournit désormais une image de build de Windows Server Core 2019. Pour plus d'informations, consultez les <a href="#">images Docker fournies par CodeBuild</a> .                     | 20 juillet 2020 |

|                                                              |                                                                                                                                                                                                                                                                                                                   |                 |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Gestionnaire de sessions</a>                     | CodeBuild vous permet désormais de suspendre une construction en cours, puis d'utiliser le gestionnaire de AWS Systems Manager session pour vous connecter au conteneur de génération et afficher l'état du conteneur . Pour plus d'informations, consultez la section <a href="#">Gestionnaire de sessions</a> . | 20 juillet 2020 |
| <a href="#">Rubrique mise à jour</a>                         | CodeBuild prend désormais en charge la spécification d'un shell à utiliser dans leurs environnements de construction dans le fichier buildspec . Pour plus d'informations, consultez la section <a href="#">Référence des spécifications de construction</a> .                                                    | 25 juin 2020    |
| <a href="#">Rapports de test avec des frameworks de test</a> | Ajout de plusieurs rubriques décrivant comment générer des rapports de CodeBuild test avec plusieurs frameworks de test. Pour de plus amples informations, veuillez consulter <a href="#">Rapports de test avec des cadres de test</a> .                                                                          | 29 mai 2020     |
| <a href="#">Sujets mis à jour</a>                            | CodeBuild prend désormais en charge l'ajout de balises aux groupes de rapports. Pour de plus amples informations, veuillez consulter <a href="#">ReportGroup</a> .                                                                                                                                                | 21 mai 2020     |

## [Support pour les rapports de test](#)

CodeBuild le support pour les rapports de test est désormais généralement disponible.

21 mai 2020

## [Sujets mis à jour](#)

CodeBuild permet désormais de créer des filtres de création de webhook pour Github et Bitbucket qui déclenche nt des builds uniquement lorsque le message de validation principal correspond à l'expression spécifiée. Pour plus d'informations, consultez les sections Exemple [GitHub de demande d'extraction et de filtre Webhook](#) et Exemple de [demande d'extraction et de filtre Webhook de Bitbucket](#).

6 mai 2020

## [Nouveaux sujets](#)

CodeBuild prend désormais en charge le partage des ressources du projet de construction et du groupe de rapports. Pour de plus amples informations, veuillez consulter [Utilisation de projets partagés](#) et [Utilisation de groupes de rapports partagés](#).

13 décembre 2019

|                                               |                                                                                                                                                                                                                                                                                                                                                                     |                  |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Sujets nouveaux et mis à jour</a> | CodeBuild prend désormais en charge les rapports de test pendant l'exécution d'un projet de construction. Pour plus d'informations, consultez les <a href="#">sections Utilisation des rapports de test</a> , <a href="#">Création d'un rapport de test</a> et <a href="#">Création d'un rapport de test à l'aide de l'AWS CLI exemple</a> .                        | 25 novembre 2019 |
| <a href="#">Rubrique mise à jour</a>          | CodeBuild prend désormais en charge les types d'environnement Linux GPU et Arm, ainsi que le type de 2xlarge calcul. Pour de plus amples informations, veuillez consulter <a href="#">Types de calcul de l'environnement de génération</a> .                                                                                                                        | 19 novembre 2019 |
| <a href="#">Sujets mis à jour</a>             | CodeBuild prend désormais en charge les numéros de version sur toutes les versions, l'exportation de variables d'environnement et AWS Secrets Manager l'intégration. Pour de plus amples informations, veuillez consulter <a href="#">Variables exportées</a> et <a href="#">Secrets Manager</a> dans la <a href="#">Syntaxe des spécifications de génération</a> . | 6 novembre 2019  |

|                                   |                                                                                                                                                                                                                                                                                                                               |                   |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Nouvelle rubrique</a> | CodeBuild prend désormais en charge les règles de notification. Vous pouvez utiliser des règles de notification pour informer les utilisateurs des modifications importantes apportées aux projets de génération. Pour de plus amples informations, veuillez consulter <a href="#">Création d'une règle de notification</a> . | 5 novembre 2019   |
| <a href="#">Sujets mis à jour</a> | CodeBuild supporte désormais les runtimes Android version 29 et Go version 1.13. Pour de plus amples informations, veuillez consulter <a href="#">Images Docker fournies par CodeBuild</a> et <a href="#">Syntaxe des spécifications de génération</a> .                                                                      | 10 septembre 2019 |
| <a href="#">Sujets mis à jour</a> | Lorsque vous créez un projet, vous pouvez désormais choisir l'image gérée par Amazon Linux 2 (AL2). Pour de plus amples informations, veuillez consulter <a href="#">Images Docker fournies par CodeBuild</a> et <a href="#">Versions d'environnement d'exécution dans l'exemple de fichier buildspec pour CodeBuild</a> .    | 16 août 2019      |



## Rubrique mise à jour

Lorsque vous créez un projet, vous pouvez désormais choisir de désactiver le chiffrement des journaux S3 et, si vous utilisez un référentiel source basé sur Git, inclure des sous-modules Git. Pour plus d'informations, voir [Créer un projet de construction dans CodeBuild](#).

8 mars 2019

## Nouvelle rubrique

CodeBuild prend désormais en charge la mise en cache locale. Vous pouvez spécifier la mise en cache locale dans un ou plusieurs des quatre modes lors de la création d'une génération.. Pour plus d'informations, consultez la section [Intégrer la mise en CodeBuild cache](#).

21 février 2019

## Nouveaux sujets

CodeBuild prend désormais en charge les groupes de filtres Webhook pour spécifier les événements qui déclenchent une compilation. Pour plus d'informations, consultez [Filtrer les événements GitHub webhook et Filtrer les événements webhook Bitbucket](#).

8 février 2019

## Nouvelle rubrique

Le guide de CodeBuild l'utilisateur indique désormais comment utiliser CodeBuild un serveur proxy. Pour plus d'informations, consultez la section [Utilisation CodeBuild avec un serveur proxy](#).

4 février 2019

## Sujets mis à jour

CodeBuild prend désormais en charge l'utilisation d'une image Amazon ECR enregistrée dans un autre AWS compte. Plusieurs rubriques ont été mises à jour pour refléter cette modification, notamment [l'exemple Amazon ECR pour CodeBuild](#), [Create a build project](#) et [Create a CodeBuild service role](#).

24 janvier 2019

## Support pour les registres Docker privés

CodeBuild prend désormais en charge l'utilisation d'une image Docker stockée dans un registre privé comme environnement d'exécution. Pour plus d'informations, voir [Registre privé avec AWS Secrets Manager échantillon](#).

24 janvier 2019

|                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                            |                  |
|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Rubrique mise à jour</a>                                           | CodeBuild prend désormais en charge l'utilisation d'un jeton d'accès pour se connecter aux référentiels GitHub (avec un jeton d'accès personnel) et Bitbucket (avec un mot de passe d'application). Pour de plus amples informations, veuillez consulter <a href="#">Création d'un projet de génération (console)</a> et <a href="#">Utilisation des jetons d'accès avec votre fournisseur de source</a> . | 6 décembre 2018  |
| <a href="#">Rubrique mise à jour</a>                                           | CodeBuild prend désormais en charge de nouvelles métriques de construction qui mesurent la durée de chaque phase d'une construction. Pour plus d'informations, consultez la section <a href="#">CodeBuild CloudWatch Mesures</a> .                                                                                                                                                                         | 15 novembre 2018 |
| <a href="#">Rubrique relative à la politique des points de terminaison VPC</a> | Les points de terminaison Amazon VPC prennent CodeBuild actuellement en charge les politiques. Pour de plus amples informations, veuillez consulter <a href="#">Création d'une stratégie de point de terminaison de VPC pour CodeBuild</a> .                                                                                                                                                               | 9 novembre 2018  |
| <a href="#">Contenu mis à jour</a>                                             | Les rubriques ont été mises à jour pour refléter cette nouvelle console.                                                                                                                                                                                                                                                                                                                                   | 30 octobre 2018  |

### [Exemple Amazon EFS](#)

CodeBuild peut monter un système de fichiers Amazon EFS lors d'une construction à l'aide des commandes du fichier buildspec d'un projet. Pour plus d'informations, consultez l'[exemple Amazon EFS pour CodeBuild](#).

26 octobre 2018

### [Webhooks Bitbucket](#)

CodeBuild supporte désormais les webhooks lorsque vous utilisez Bitbucket pour votre dépôt. Pour de plus amples informations, veuillez consulter [Exemple de demande d'extraction Bitbucket pour CodeBuild](#).

2 octobre 2018

### [Journaux S3](#)

CodeBuild prend désormais en charge les journaux de génération dans un compartiment S3. Auparavant, vous ne pouviez créer des journaux qu'à l'aide de CloudWatch journaux. Pour de plus amples informations, veuillez consulter [Création d'un projet](#).

17 septembre 2018

## [Sources d'entrée multiples et artefacts de sortie multiples](#)

CodeBuild prend désormais en charge les projets qui utilisent plusieurs sources d'entrée et publient plusieurs ensembles d'artefacts. Pour plus [d'informations, voir Échantillonnage et CodePipeline intégration de plusieurs sources d'entrée et d' CodeBuild artefacts d'entrée et échantillon d'artefacts de sortie multiples](#).

30 août 2018

## [Exemple de versionnement sémantique](#)

Le guide de CodeBuild l'utilisateur contient désormais un exemple basé sur des cas d'utilisation qui montre comment utiliser le versionnement sémantique pour créer des noms d'artefacts au moment de la création. Pour de plus amples informations, veuillez consulter [Exemple de gestion sémantique des versions de vos artefacts de génération](#).

14 août 2018

## [Nouvel exemple de site Web statique](#)

Le guide de CodeBuild l'utilisateur contient désormais un exemple basé sur des cas d'utilisation qui montre comment héberger la sortie de build dans un compartiment S3. L'exemple s'appuie sur la prise en charge récente des artefacts de génération non chiffrés. Pour de plus amples informations, veuillez consulter [Création d'un site web statique avec la sortie de génération hébergée dans un compartiment Amazon S3](#).

14 août 2018

### [Support pour remplacer le nom d'un artefact par le versionnement sémantique](#)

Vous pouvez désormais utiliser le versionnement sémantique pour spécifier un format CodeBuild utilisé pour nommer les artefacts de construction. Cela peut s'avérer utile, car un artefact de génération avec un nom codé de manière fixe remplacera les artefacts de génération précédents qui utilisent le même nom codé de manière fixe. Par exemple, si une génération est déclenchée plusieurs fois par jour, vous pouvez désormais ajouter un horodatage à son nom d'artefact. Chaque nom d'artefact de génération est unique, ce qui évite d'écraser les artefacts des générations précédentes.

7 août 2018

### [Support des artefacts de construction non chiffrés](#)

CodeBuild prend désormais en charge les builds avec des artefacts de build non chiffrés. Pour plus d'informations, voir [Création d'un projet de génération \(console\)](#).

26 juillet 2018

### [Support pour les CloudWatch métriques et les alarmes Amazon](#)

CodeBuild fournit désormais une intégration avec CloudWatch les métriques et les alarmes. Vous pouvez utiliser la CloudWatch console CodeBuild ou pour surveiller les builds au niveau du projet et du compte. Pour de plus amples informations, veuillez consulter [Surveillance des générations](#).

19 juillet 2018

### [Support pour signaler l'état d'une construction](#)

CodeBuild peut désormais signaler l'état du début et de la fin d'une compilation à votre fournisseur source. Pour plus d'informations, voir [Créer un projet de construction dans CodeBuild](#).

10 juillet 2018

### [Variables d'environnement ajoutées à CodeBuild la documentation](#)

La page [Variables d'environnement dans les environnements de génération](#) a été mise à jour avec les variables d'environnement CODEBUILD\_BUILD\_ID, CODEBUILD\_LOG\_PATH et CODEBUILD\_START\_TIME.

9 juillet 2018



### [Support pour un finally bloc dans le fichier buildspec](#)

La CodeBuild documentation a été mise à jour avec des détails sur le finally bloc facultatif dans un fichier buildspec. Les commandes du bloc final s'exécutent toujours après les commandes du bloc de commandes correspondant. Pour de plus amples informations, veuillez consulter [Syntaxe des spécifications de génération](#).

le 20 juin 2018

### [CodeBuild notifications de mise à jour des agents](#)

La CodeBuild documentation a été mise à jour avec des informations détaillées sur la manière dont vous pouvez utiliser Amazon SNS pour être averti lorsque de nouvelles versions de l'agent CodeBuild sont publiées. Pour plus d'informations, voir [Recevoir des notifications pour les nouvelles versions de AWS CodeBuild l'agent](#).

15 juin 2018

## Mises à jour antérieures

Le tableau ci-après décrit des modifications importantes apportées dans chaque version du Guide de l'utilisateur AWS CodeBuild avant juin 2018.

| Modifier                           | Description                                                                    | Date        |
|------------------------------------|--------------------------------------------------------------------------------|-------------|
| Prise en charge des builds Windows | CodeBuild prend désormais en charge les versions pour la plate-forme Microsoft | 25 mai 2018 |

| Modifier                                                                                       | Description                                                                                                                                                                                                                                                                            | Date        |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
|                                                                                                | Windows Server, y compris un environnement de génération préconfiguré pour .NET Core 2.0 sous Windows. Pour plus d'informations, consultez <a href="#">Exécutez des exemples Microsoft Windows pour CodeBuild</a> .                                                                    |             |
| Prise en charge de l'idempotence des générations                                               | Lorsque vous exécutez la <code>start-build</code> commande avec le AWS Command Line Interface (AWS CLI), vous pouvez spécifier que le build est idempotent. Pour plus d'informations, consultez <a href="#">Exécution d'une génération (AWS CLI)</a> .                                 | 15 mai 2018 |
| Prise en charge du remplacement d'un plus grand nombre de paramètres d'un projet de génération | Vous pouvez désormais remplacer plusieurs paramètres d'un projet de génération lorsque vous créez une génération. Les remplacements concernent alors uniquement cette génération. Pour plus d'informations, consultez <a href="#">Exécuter AWS CodeBuild les builds manuellement</a> . | 15 mai 2018 |

| Modifier                                         | Description                                                                                                                                                                                                                                                                                                                           | Date         |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Prise en charge des points de terminaison VPC    | Vous pouvez désormais utiliser les points de terminaison d'un VPC pour améliorer la sécurité de vos générations. Pour plus d'informations, consultez <a href="#">Utilisation de points de terminaison de VPC</a> .                                                                                                                    | 18 mars 2018 |
| Prise en charge des déclencheurs                 | Vous pouvez désormais créer des déclencheurs afin de programmer des générations à intervalles réguliers. Pour plus d'informations, consultez <a href="#">Création de AWS CodeBuild déclencheurs</a> .                                                                                                                                 | 28 mars 2018 |
| Documentation sur les points de terminaison FIPS | Vous pouvez maintenant apprendre à utiliser le AWS Command Line Interface (AWS CLI) ou un AWS SDK pour savoir comment utiliser l'un des quatre points CodeBuild de terminaison FIPS (Federal Information Processing Standards). Pour plus d'informations, consultez <a href="#">Spécifiez le AWS CodeBuild point de terminaison</a> . | 28 mars 2018 |

| Modifier                                                                                                 | Description                                                                                                                                                                                                                                         | Date             |
|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AWS CodeBuild disponible en Asie-Pacifique (Mumbai), en Europe (Paris) et en Amérique du Sud (São Paulo) | AWS CodeBuild est désormais disponible dans les régions Asie-Pacifique (Mumbai), Europe (Paris) et Amérique du Sud (São Paulo). Pour plus d'informations, consultez <a href="#">AWS CodeBuild</a> dans le Référence générale d'Amazon Web Services. | 28 mars 2018     |
| GitHub Support pour serveurs d'entreprise                                                                | CodeBuild peut désormais créer à partir du code source stocké dans un référentiel GitHub Enterprise Server. Pour plus d'informations, consultez <a href="#">Exécutez l'exemple de serveur d' GitHub entrepris e.</a>                                | 25 janvier 2018  |
| Prise en charge de la profondeur du clone Git                                                            | CodeBuild prend désormais en charge la création d'un clone superficiel avec un historique tronqué au nombre de validations spécifié. Pour plus d'informations, consultez <a href="#">Création d'un projet de génération.</a>                        | 25 janvier 2018  |
| Prise en charge de VPC                                                                                   | Les générations activées pour VPC sont désormais en mesure d'accéder aux ressources à l'intérieur de votre VPC. Pour plus d'informations, consultez <a href="#">Prise en charge de VPC.</a>                                                         | 27 novembre 2017 |

| Modifier                                            | Description                                                                                                                                                                                                                                                                                   | Date             |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Prise en charge de la mise en cache des dépendances | CodeBuild prend désormais en charge la mise en cache des dépendances. Cela permet d' CodeBuild enregistrer certaines parties réutilisables de l'environnement de construction dans le cache et de les utiliser entre les versions.                                                            | 27 novembre 2017 |
| Prise en charge des badges de génération            | CodeBuild prend désormais en charge l'utilisation de badges de construction, qui fournissent une image intégrée et générée dynamiquement (badge) qui affiche l'état de la dernière version d'un projet. Pour plus d'informations, consultez <a href="#">Exemple de badges de génération</a> . | 27 novembre 2017 |
| AWS Config intégration                              | AWS Config prend désormais en charge en CodeBuild tant que AWS ressource, ce qui signifie que le service peut suivre vos CodeBuild projets. Pour plus d'informations sur AWS Config, voir <a href="#">AWS Config échantillon</a> .                                                            | 20 octobre 2017  |

| Modifier                                                                            | Description                                                                                                                                                                                                                                                                                                                    | Date              |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Reconstruire automatiquement le code source mis à jour dans les GitHub référentiels | Si votre code source est stocké dans un GitHub référentiel, vous pouvez activer la reconstruction AWS CodeBuild de votre code source chaque fois qu'une modification de code est envoyée au référentiel. Pour plus d'informations, consultez <a href="#">Exécutez l' GitHub exemple de pull request et de filtre Webhook</a> . | 21 septembre 2017 |

| Modifier                                                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Date              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Nouvelles méthodes de stockage et de récupération de variables d'environnement sensibles ou volumineuses dans Amazon EC2 Systems Manager Parameter Store | <p>Vous pouvez désormais utiliser la AWS CodeBuild console ou le AWS CLI pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans Amazon EC2 Systems Manager Parameter Store. Vous pouvez également utiliser la AWS CodeBuild console pour stocker ces types de variables d'environnement dans Amazon EC2 Systems Manager Parameter Store. Auparavant, vous pouviez uniquement récupérer ces types de variables d'environnement en les incluant dans une spécification de génération ou en exécutant des commandes de génération pour automatiser l' AWS CLI. Vous ne pouviez stocker ces types de variables d'environnement qu'à l'aide de la console Amazon EC2 Systems Manager Parameter Store. Pour plus d'informations, reportez-vous aux <a href="#">Modifier les paramètres du projet de construction</a> sections <a href="#">Création d'un projet de génération</a>, et <a href="#">Exécuter les builds manuellement</a>.</p> | 14 septembre 2017 |

| Modifier                                                                                                                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Date            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Prise en charge de la suppression de génération                                                                                                                                      | Vous pouvez désormais supprimer des générations dans AWS CodeBuild. Pour plus d'informations, consultez <a href="#">Suppression de générations</a> .                                                                                                                                                                                                                                                                                                                                                         | 31 août 2017    |
| Méthode mise à jour pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans le magasin de paramètres Amazon EC2 Systems Manager à l'aide d'un buildspec | AWS CodeBuild facilite désormais l'utilisation d'une spécification de construction pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans le magasin de paramètres Amazon EC2 Systems Manager. Auparavant, vous pouviez uniquement récupérer ces types de variables d'environnement en exécutant des commandes de génération pour automatiser l' AWS CLI. Pour plus d'informations, consultez le parameter-store mappage dans <a href="#">Syntaxe d'un fichier buildspec</a> . | le 10 août 2017 |
| AWS CodeBuild prend en charge Bitbucket                                                                                                                                              | CodeBuild peut désormais être compilé à partir du code source stocké dans un dépôt Bitbucket. Pour plus d'informations, consultez <a href="#">Création d'un projet de génération</a> et <a href="#">Exécuter les builds manuellement</a> .                                                                                                                                                                                                                                                                   | le 10 août 2017 |



| Modifier                                                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                         | Date         |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| AWS CodeBuild disponible dans l'ouest des États-Unis (Californie du Nord), en Europe (Londres) et au Canada (centre) | AWS CodeBuild est désormais disponible dans les régions de l'ouest des États-Unis (Californie du Nord), de l'Europe (Londres) et du Canada (centre). Pour plus d'informations, consultez <a href="#">AWS CodeBuild</a> dans le Référence générale d'Amazon Web Services.                                                                                                                            | 29 juin 2017 |
| Autres noms et emplacements de fichiers de spécification de génération pris en charge                                | Vous pouvez désormais spécifier un autre nom de fichier ou emplacement pour un fichier de spécification de génération à utiliser pour un projet de génération au lieu d'un fichier de spécification de génération par défaut nommé <code>buildspec.yml</code> à la racine du code source. Pour plus d'informations, consultez <a href="#">Nom de fichier buildspec et emplacement de stockage</a> . | 27 juin 2017 |

| Modifier                                           | Description                                                                                                                                                                                                                                                                                       | Date         |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Exemple de notifications de génération mis à jour  | CodeBuild fournit désormais un support intégré pour les notifications de build via Amazon CloudWatch Events et Amazon Simple Notification Service (Amazon SNS). L'exemple <a href="#">Exemple de notifications de génération</a> précédent a été mis à jour pour montrer ce nouveau comportement. | 22 juin 2017 |
| Exemple Docker dans une image personnalisée ajouté | Un exemple montrant comment utiliser CodeBuild une image Docker personnalisée pour créer et exécuter une image Docker a été ajouté. Pour plus d'informations, consultez le <a href="#">Exemple Docker dans une image personnalisée</a> .                                                          | 7 juin 2017  |

| Modifier                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Date        |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| Récupère le code source pour les GitHub pull requests | Lorsque vous exécutez un build CodeBuild qui repose sur le code source stocké dans un GitHub référentiel, vous pouvez désormais spécifier un ID de GitHub pull request à compiler. Vous pouvez également spécifier plutôt un ID de validation, un nom de branche ou un nom de balise. Pour plus d'informations, voir la valeur de la version source dans <a href="#">Exécution d'une génération (console)</a> ou la sourceVersion valeur dans <a href="#">Exécution d'une génération (AWS CLI)</a> . | 6 juin 2017 |

| Modifier                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Date       |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Version de spécification de génération mise à jour        | Une nouvelle version du format de spécification de génération a été publiée. La version 0.2 résout le problème de l' CodeBuild exécution de chaque commande de construction dans une instance distincte du shell par défaut. En outre, dans la version 0.2, <code>environment_variables</code> est renommé en <code>env</code> et <code>plaintext</code> est renommé en <code>variables</code> . Pour plus d'informations, consultez <a href="#">Référence de spécification de construction pour CodeBuild</a> . | 9 mai 2017 |
| Dockerfiles pour créer des images disponibles dans GitHub | Les définitions de nombreuses images de build fournies par AWS CodeBuild sont disponibles sous forme de Dockerfiles dans GitHub. Pour plus d'informations, consultez la colonne Définition du tableau dans <a href="#">Images Docker fournies par CodeBuild</a> .                                                                                                                                                                                                                                                | 2 mai 2017 |

| Modifier                                                                                                                                                       | Description                                                                                                                                                                                                                                                                                                                                                   | Date           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| AWS CodeBuild disponible en Europe (Francfort), en Asie-Pacifique (Singapour), en Asie-Pacifique (Sydney) et en Asie-Pacifique (Tokyo)                         | AWS CodeBuild est désormais disponible dans les régions Europe (Francfort), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney) et Asie-Pacifique (Tokyo). Pour plus d'informations, consultez <a href="#">AWS CodeBuild</a> dans le Référence générale d'Amazon Web Services.                                                                                | 21 mars 2017   |
| CodePipeline support aux actions de test pour CodeBuild                                                                                                        | Vous pouvez désormais ajouter des éléments à un pipeline dans le cadre CodePipeline d'une action de test qui utilise CodeBuild . Pour plus d'informations, consultez <a href="#">Ajouter une action de CodeBuild test à un pipeline (CodePipeline console)</a> .                                                                                              | 8 mars 2017    |
| Les fichiers de spécification de génération prennent en charge l'extraction de la sortie de génération depuis des répertoires de niveau supérieur sélectionnés | Les fichiers Buildspec vous permettent désormais de spécifier des répertoires individuels de premier niveau dont vous pouvez demander l'inclusion dans les artefacts de sortie CodeBuild de construction. Pour ce faire, vous devez utiliser le mappage base-directory . Pour plus d'informations, consultez <a href="#">Syntaxe d'un fichier buildspec</a> . | 8 février 2017 |

| Modifier                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Date            |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Variables d'environnement intégrées                       | AWS CodeBuild fournit des variables d'environnement intégrées supplémentaires que vos builds peuvent utiliser. Il s'agit notamment de variables d'environnement qui décrivent l'entité qui a démarré la génération, l'URL vers le référentiel du code source, l'ID de version du code source, etc. Pour plus d'informations, consultez <a href="#">Variables d'environnement dans les environnements de génération</a> . | 30 janvier 2017 |
| AWS CodeBuild disponible dans l'est des États-Unis (Ohio) | AWS CodeBuild est désormais disponible dans la région USA Est (Ohio). Pour plus d'informations, consultez <a href="#">AWS CodeBuild</a> dans le Référence générale d'Amazon Web Services.                                                                                                                                                                                                                                | 19 janvier 2017 |

| Modifier                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                   | Date            |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Informations sur les comportements de shell et de commande | CodeBuild exécute chaque commande que vous spécifiez dans une instance distincte du shell par défaut d'un environnement de génération. Ce comportement par défaut peut produire des effets inattendus sur vos commandes. Nous vous recommandons certaines approches pour contourner ce comportement par défaut si nécessaire. Pour plus d'informations, consultez <a href="#">Shells et commandes dans les environnements de génération</a> . | 9 décembre 2016 |
| Informations sur les variables d'environnement             | CodeBuild fournit plusieurs variables d'environnement que vous pouvez utiliser dans vos commandes de compilation. Vous pouvez également définir vos propres variables d'environnement. Pour plus d'informations, consultez <a href="#">Variables d'environnement dans les environnements de génération</a> .                                                                                                                                  | 7 décembre 2016 |
| Rubrique de dépannage                                      | Des informations de dépannage sont désormais disponibles. Pour plus d'informations, consultez <a href="#">Résolution des problèmes AWS CodeBuild</a> .                                                                                                                                                                                                                                                                                        | 5 décembre 2016 |

| Modifier                                   | Description                                                                                                                                              | Date              |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Version initiale du plug-in Jenkins        | Il s'agit de la version initiale du plugin CodeBuild Jenkins. Pour plus d'informations, consultez <a href="#">À utiliser AWS CodeBuild avec Jenkins.</a> | 5 décembre 2016   |
| Version initiale du Guide de l'utilisateur | Il s'agit de la première édition du guide de l'utilisateur CodeBuild .                                                                                   | 1er décembre 2016 |



Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.