

Guide de l'utilisateur

AWS CodeDeploy



Version de l'API 2014-10-06

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

AWS CodeDeploy: 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 CodeDeploy ?	1
Avantages de AWS CodeDeploy	2
Vue d'ensemble des plateformes CodeDeploy informatiques	3
Vue d'ensemble des types de CodeDeploy déploiement	11
Vue d'ensemble d'un déploiement sur place	13
Vue d'ensemble d'un blue/green déploiement	14
Nous voulons entendre parler de vous	19
Composants principaux	19
Application	19
Plateforme de calcul	20
Configuration de déploiement	21
Groupe de déploiement	21
Type de déploiement	22
Type de déploiement	23
Revision	23
Rôle de service	24
Révision cible	24
Autres composants	24
Déploiements	25
Déploiements sur une plateforme de calcul AWS Lambda	25
Déploiements sur une plateforme de calcul Amazon ECS	29
Déploiements sur une plateforme informatique EC2/sur site	42
Fichiers de spécifications des applications	49
AppSpec fichiers sur une plateforme de calcul Amazon ECS	50
AppSpec fichiers sur une plate-forme AWS Lambda informatique	50
AppSpec fichiers sur une plate-forme informatique EC2/sur site	50
Comment l' CodeDeploy agent utilise le AppSpec fichier	51
Prise en main	52
Étape 1 : Configuration	52
Inscrivez-vous pour un Compte AWS	52
Création d'un utilisateur doté d'un accès administratif	53
Octroi d'un accès par programmation	54
Étape 2 : créer un rôle de service	57
Création d'un rôle de service (console)	59

Création d'un rôle de service (CLI)	62
Obtenir l'ARN du rôle de service (console)	65
Obtenez l'ARN du rôle de service (CLI)	65
Étape 3 : Limiter les autorisations de l' CodeDeploy utilisateur	65
Étape 4 : Création d'un profil d'instance IAM	70
Créez un profil d'instance IAM pour vos instances Amazon EC2 (CLI)	70
Création d'un profil d'instance IAM pour vos instances Amazon EC2 (console)	75
Intégrations de produits et services	79
Intégration avec d'autres AWS services	79
Amazon EC2 Auto Scaling	88
Elastic Load Balancing	97
Intégration avec les produits et services des partenaires	102
GitHub	108
Exemples d'intégration issus de la communauté	112
Billets de blogs	112
Didacticiels	114
Tutoriel : Déploiement WordPress sur une instance autre que Windows	114
Étape 1 : lancer une instance Amazon EC2	115
Étape 2 : configurer votre contenu source	118
Étape 3 : Chargez votre application sur Amazon S3	123
Étape 4 : Déployez votre application	128
Étape 5 : Mettez à jour et redéployez votre application	135
Étape 6 : Nettoyer	140
Tutoriel : Déployer une application Hello World sur une instance Windows Server	143
Étape 1 : lancer une instance Amazon EC2	144
Étape 2 : configurer votre contenu source	146
Étape 3 : Chargez votre application sur Amazon S3	150
Étape 4 : Déployez votre application	155
Étape 5 : Mettez à jour et redéployez votre application	160
Étape 6 : Nettoyer	164
Tutoriel : Déployer une application sur une instance locale	167
Conditions préalables	168
Étape 1 : Configuration de l'instance sur site	168
Étape 2 : Création d'un exemple de révision d'application	169
Étape 3 : Regroupez et chargez la révision de votre application sur Amazon S3	174
Étape 4 : Déployez la révision de votre application	175

Étape 5 : vérifier votre déploiement	175
Étape 6 : nettoyer les ressources	175
Tutoriel : Déploiement vers un groupe Auto Scaling	177
Conditions préalables	178
Étape 1 : Création et configuration du groupe Auto Scaling	179
Étape 2 : Déployer l'application dans le groupe Auto Scaling	185
Étape 3 : Vérifiez vos résultats	195
Étape 4 : augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling	197
Étape 5 : Vérifiez à nouveau vos résultats	198
Étape 6 : Nettoyer	201
Tutoriel : Déployer une application depuis GitHub	203
Conditions préalables	204
Étape 1 : créer un GitHub compte	204
Étape 2 : Création d'un GitHub référentiel	205
Étape 3 : télécharger un exemple d'application dans votre GitHub référentiel	207
Étape 4 : provisionner une instance	212
Étape 5 : Création d'une application et d'un groupe de déploiement	213
Étape 6 : Déployer l'application sur l'instance	215
Étape 7 : Surveiller et vérifier le déploiement	219
Étape 8 : nettoyer	221
Tutoriel : Déployer une application dans Amazon ECS	223
Conditions préalables	225
Étape 1 : mettez à jour votre application Amazon ECS	226
Étape 2 : Création du AppSpec fichier	227
Étape 3 : utiliser la CodeDeploy console pour déployer votre application	228
Étape 4 : nettoyer	233
Tutoriel : Déployer un service Amazon ECS avec un test de validation	234
Conditions préalables	237
Étape 1 : créer un écouteur de test	237
Étape 2 : mettez à jour votre application Amazon ECS	238
Étape 3 : Création d'une fonction Lambda Lifecycle Hook	238
Étape 4 : mettez à jour votre AppSpec fichier	241
Étape 5 : utiliser la CodeDeploy console pour déployer votre service Amazon ECS	242
Étape 6 : Afficher la sortie de votre fonction Lambda hook dans Logs CloudWatch	244
Étape 7 : nettoyer	245
Tutoriel : Déployer une fonction Lambda à l'aide de SAM AWS	246

Conditions préalables	247
Étape 1 : configurer votre infrastructure	247
Étape 2 : mise à jour de la fonction Lambda	263
Étape 3 : Déploiement de la fonction Lambda mise à jour	265
Étape 4 : Afficher les résultats de votre déploiement	268
Étape 5 : nettoyer	271
Travailler avec l' CodeDeploy agent	273
Systèmes d'exploitation pris en charge par l' CodeDeploy agent	274
Systèmes d'exploitation Amazon EC2 AML pris en charge	274
Systèmes d'exploitation sur site pris en charge	274
Protocole de communication et port pour l' CodeDeploy agent	274
Historique des versions de l' CodeDeploy agent	275
Gérer le CodeDeploy processus	290
Révision de l'application et nettoyage des fichiers journaux	291
Fichiers installés par l' CodeDeploy agent	291
Gestion des opérations des CodeDeploy agents	295
Vérifiez que l' CodeDeploy agent est en cours d'exécution	295
Déterminer la version de l' CodeDeploy agent	297
Installation de l' CodeDeploy agent	299
Mettre à jour l' CodeDeploy agent	311
Désinstallez l' CodeDeploy agent	315
Envoyer les journaux des CodeDeploy agents à CloudWatch	316
Utilisation d'instances	321
Comparaison des instances Amazon EC2 avec des instances sur site	321
Tâches d'instance pour CodeDeploy	323
Marquage des instances pour les déploiements CodeDeploy	325
Exemple 1 : groupe de balises unique, étiquette unique	326
Exemple 2 : groupe de balises unique, balises multiples	327
Exemple 3 : plusieurs groupes de balises, balises uniques	329
Exemple 4 : plusieurs groupes de balises, plusieurs balises	331
Utilisation des instances Amazon EC2	335
Créez une instance Amazon EC2 pour CodeDeploy	335
Création d'une instance Amazon EC2 (modèle)CloudFormation	342
Configuration d'une instance Amazon EC2	353
Gestion des instances sur site	357
Conditions préalables à la configuration d'une instance sur site	358

Inscrire une instance sur site	360
Gestion des opérations des instances sur site	398
Afficher les détails de l'instance	406
Afficher les détails de l'instance (console)	406
Afficher les détails de l'instance (CLI)	407
État de l'instance	407
État de santé	408
À propos du nombre minimum d'instances saines	410
À propos du nombre minimum d'instances saines par zone de disponibilité	414
Utilisation des configurations de déploiement	417
Configurations de déploiement sur une plate-forme informatique EC2/sur site	417
Configurations de déploiement prédéfinies	417
Configurations de déploiement sur une plateforme de calcul Amazon ECS	422
Configurations de déploiement prédéfinies pour Amazon ECS	422
Configurations de déploiement pour les déploiements CloudFormation bleu/vert (Amazon ECS)	423
Configurations de déploiement sur une plateforme AWS Lambda informatique	424
Configurations de déploiement prédéfinies pour Lambda	424
Création d'une configuration de déploiement	426
Création d'une configuration de déploiement (console)	426
Création d'une configuration de déploiement (AWS CLI)	429
Afficher les détails de configuration du déploiement	430
Afficher les détails de configuration du déploiement (console)	430
Afficher la configuration de déploiement (CLI)	431
Supprimer une configuration de déploiement	431
Utilisation d'applications	433
Création d'une application	434
Création d'une application pour un déploiement sur place (console)	436
Création d'une application pour un blue/green déploiement (console)	439
Création d'une application pour le déploiement d'un service Amazon ECS (console)	445
Création d'une application pour le déploiement d'une AWS Lambda fonction (console)	448
Création d'une application (CLI)	450
Afficher les détails de l'application	450
Afficher les détails de l'application (console)	450
Afficher les détails de l'application (CLI)	451
Création d'une règle de notification	451

Renommer une application	454
Supprimer une application	455
Supprimer une application (console)	455
Supprimer une application (AWS CLI)	456
Utilisation des groupes de déploiement	457
Groupes de déploiement dans des déploiements de plateforme de calcul Amazon ECS	457
Groupes de déploiement dans les déploiements de plateformes AWS Lambda informatiques ..	457
Groupes de déploiement dans les déploiements de plateformes de calcul EC2/sur site	457
Création d'un groupe de déploiement	459
Création d'un groupe de déploiement pour un déploiement sur place (console)	459
Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site (console)	464
Création d'un groupe de déploiement pour un déploiement Amazon ECS (console)	471
Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2	473
Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS	474
Création d'un groupe de déploiement (CLI)	479
Afficher les détails du groupe de déploiement	481
Afficher les détails du groupe de déploiement (console)	481
Afficher les détails du groupe de déploiement (CLI)	482
Modifier les paramètres du groupe de déploiement	482
Modifier les paramètres du groupe de déploiement (console)	483
Modifier les paramètres du groupe de déploiement (CLI)	484
Configuration des options avancées d'un groupe de déploiement	485
Supprimer un groupe de déploiement	489
Supprimer un groupe de déploiement (console)	489
Supprimer un groupe de déploiement (CLI)	490
Utilisation des révisions d'application	491
Planifier une révision	491
Ajouter un AppSpec fichier	492
Ajouter un AppSpec fichier pour un déploiement Amazon ECS	492
Ajouter un AppSpec fichier pour un déploiement AWS Lambda	496
Ajouter un AppSpec fichier pour un déploiement EC2/sur site	498
Choisissez un type de référentiel	502
Propulser une révision	505

Envoyez une révision à l'aide du AWS CLI	508
Afficher les détails de révision de l'application	510
Afficher les détails de révision de l'application (console)	510
Afficher les détails de révision de l'application (CLI)	511
Enregistrer une révision d'application	512
Enregistrez une révision dans Amazon S3 avec CodeDeploy (CLI)	512
Enregistrer une révision GitHub avec CodeDeploy (CLI)	514
Utilisation de déploiements	515
Création d'un déploiement	516
Conditions préalables au déploiement	517
Création d'un déploiement d'Amazon ECS Compute Platform (console)	520
Création d'un déploiement de plateforme de calcul AWS Lambda (console)	522
Création d'un déploiement de plate-forme de calcul EC2/sur site (console)	524
Création d'un déploiement de plate-forme de calcul Amazon ECS (CLI)	529
Création d'un déploiement de plateforme de calcul AWS Lambda (CLI)	531
Création d'un déploiement de plate-forme de calcul EC2/sur site (CLI)	532
Créez un blue/green déploiement Amazon ECS via CloudFormation	536
Afficher les détails du déploiement	540
Afficher les détails du déploiement (console)	541
Afficher les détails du déploiement (CLI)	542
Afficher les données du journal de déploiement	543
Afficher les données du fichier journal dans la CloudWatch console Amazon	543
Afficher les fichiers journaux d'une instance	543
Arrêter un déploiement	547
Arrêt d'un déploiement (console)	547
Arrêt d'un déploiement (interface CLI)	548
Redéploiement et annulation d'un déploiement	548
Annulations automatiques	549
Annulations manuelles	549
Flux de travail de restauration et de redéploiement	550
Comportement d'annulation avec du contenu existant	551
Déployer une application dans un autre AWS compte	555
Étape 1 : créer un compartiment S3 dans l'un ou l'autre des comptes	556
Étape 2 : accorder des autorisations de compartiment Amazon S3 au profil d'instance IAM du compte de production	556
Étape 3 : Création de ressources et d'un rôle multicompte dans le compte de production	557

Étape 4 : télécharger la révision de l'application dans le compartiment Amazon S3	559
Étape 5 : assumer le rôle multi-comptes et déployer des applications	559
Valider un package de déploiement sur une machine locale	560
Conditions préalables	560
Création d'un déploiement local	563
Exemples	566
Surveillance des déploiements	568
Outils automatisés	569
Outils manuels	570
Surveillance des déploiements à l'aide des outils Amazon CloudWatch	571
Surveillance des déploiements à l'aide CloudWatch d'alarmes	572
Surveillance des déploiements avec Amazon CloudWatch Events	573
Surveillance des déploiements avec AWS CloudTrail	576
CodeDeploy informations dans CloudTrail	577
Comprendre les entrées du fichier CodeDeploy journal	577
Surveillance des déploiements à l'aide des notifications d'événements Amazon SNS	579
Accorder des autorisations Amazon SNS à un rôle de service	580
Création d'un déclencheur pour un CodeDeploy événement	581
Modifier un déclencheur dans un groupe de déploiement	589
Supprimer un déclencheur d'un groupe de déploiement	591
Formats de données JSON pour les déclencheurs	592
Sécurité	595
Protection des données	596
Confidentialité du trafic inter-réseau	597
Chiffrement au repos	597
Chiffrement en transit	598
Gestion des clés de chiffrement	598
Gestion des identités et des accès	598
Public ciblé	598
Authentification par des identités	599
Gestion de l'accès à l'aide de politiques	600
Comment AWS CodeDeploy fonctionne avec IAM	601
AWS politiques gérées (prédéfinies) pour CodeDeploy	607
CodeDeploy mises à jour des politiques AWS gérées	615
Exemples de politiques basées sur l'identité	620
Résolution des problèmes	628

CodeDeploy référence aux autorisations	629
Prévention du cas de figure de l'adjoint désorienté entre services	638
Intervention en cas d'incidents	641
Audit de toutes les interactions avec CodeDeploy	641
Alertes et gestion des incidents	641
Validation de conformité	642
Résilience	642
Sécurité de l'infrastructure	643
Référence	644
AppSpec référence de fichier	644
AppSpec fichiers sur une plateforme de calcul Amazon ECS	645
AppSpec fichiers sur une plate-forme AWS Lambda informatique	645
AppSpec fichiers sur une plate-forme informatique EC2/sur site	645
AppSpec Structure du fichier	646
AppSpec Exemple de fichier	694
AppSpec Espacement des fichiers	700
Validez votre AppSpec fichier et son emplacement	702
Référence de configuration de l'agent	702
Rubriques en relation	707
CloudFormation référence du modèle	708
Utilisation CodeDeploy avec Amazon Virtual Private Cloud	711
Disponibilité	712
Créez des points de terminaison VPC pour CodeDeploy	715
Configuration de l' CodeDeploy agent et des autorisations IAM	716
Référence du kit de ressources	716
Noms des compartiments du kit de ressources par région	717
Contenu du kit de ressources	719
Afficher la liste des fichiers du kit de ressources	721
Téléchargez les fichiers du kit de ressources	722
Quotas	725
Résolution des problèmes	732
Dépannage des problèmes généraux	732
Liste de contrôle de dépannage général	733
CodeDeploy les ressources de déploiement ne sont prises en charge que dans certaines AWS régions	734

Les procédures décrites dans ce guide ne correspondent pas à celles de la CodeDeploy console	735
Les rôles IAM requis ne sont pas disponibles	735
L'utilisation de certains éditeurs de texte pour créer des AppSpec fichiers et des scripts shell peut entraîner l'échec des déploiements.	736
L'utilisation du Finder sur MacOS pour grouper une révision d'application peut entraîner l'échec des déploiements	737
Résoudre les problèmes de déploiement d'EC2/sur site	737
CodeDeploy erreur d'identification CommandPoller manquante dans le plugin	738
Le déploiement échoue avec le message « Échec de la validation du message PKCS7 signé »	738
Le déploiement ou redéploiement des mêmes fichiers sur les mêmes emplacements d'instance échoue avec l'erreur « Le déploiement a échoué car un fichier spécifié existe déjà à l'emplacement »	739
Les longs chemins de fichier provoquent des erreurs « Aucun fichier ou répertoire de ce type »	742
Des processus de longue durée peuvent entraîner l'échec des déploiements	743
Résolution des problèmes liés à un échec AllowTraffic du cycle de vie sans qu'aucune erreur ne soit signalée dans les journaux de déploiement	744
Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic	745
Résolution des problèmes liés à un échec du cycle de vie d'un DownloadBundle déploiement avec UnknownError : non ouvert pour lecture	746
Résolution des erreurs d'événements de cycle de vie ignorés	747
PowerShell Les scripts Windows n'utilisent pas la version 64 bits de Windows PowerShell par défaut	749
Résoudre les problèmes de déploiement d'Amazon ECS	750
Un délai d'attente survient pendant l'attente d'un ensemble de tâches de remplacement	751
Un délai d'attente survient pendant l'attente de la poursuite d'une notification	751
Le rôle IAM ne dispose pas de suffisamment d'autorisations	752
Le délai de déploiement a expiré en attendant un rappel de statut	752
Le déploiement a échoué car une ou plusieurs fonctions de validation des événements du cycle de vie ont échoué	753
L'ELB n'a pas pu être mis à jour en raison de l'erreur suivante : Le groupe cible de l'ensemble de tâches principal doit être derrière l'écouteur	754
Mon déploiement échoue parfois lorsque j'utilise Auto Scaling	755

Seul ALB prend en charge le routage progressif du trafic, utilisez plutôt AllAtOnce le routage du trafic lorsque vous êtes un groupe create/update de déploiement	756
Même si mon déploiement a réussi, l'ensemble de tâches de remplacement échoue aux tests de santé d'Elastic Load Balancing, et mon application est en panne	756
Puis-je associer plusieurs équilibres de charge à un groupe de déploiement ?	758
Puis-je effectuer des déploiements CodeDeploy bleu/vert sans équilibreur de charge ?	758
Comment puis-je mettre à jour mon service Amazon ECS avec de nouvelles informations lors d'un déploiement ?	758
Résoudre les problèmes de déploiement AWS de Lambda	759
AWS Lambda les déploiements échouent après l'arrêt manuel d'un déploiement Lambda pour lequel aucune annulation n'est configurée	759
Résolution des problèmes de groupe de déploiement	760
Le balisage d'une instance dans le cadre d'un groupe de déploiement ne déploie pas automatiquement votre application sur la nouvelle instance	760
Résolution des problèmes d'instance	760
Les balises doivent être définies correctement	760
AWS CodeDeploy l'agent doit être installé et exécuté sur les instances	761
Les déploiements n'échouent pas pendant une heure lorsqu'une instance est mise hors service au cours d'un déploiement	761
Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances	761
Créez un nouveau fichier CodeDeploy journal s'il a été supprimé accidentellement	762
Résolution des erreurs de déploiement « InvalidSignatureException — Signature expirée : [heure] est maintenant antérieure à [heure] »	762
Résoudre les problèmes liés aux GitHub jetons	763
GitHub OAuth Jeton non valide	763
Nombre maximum de GitHub OAuth jetons dépassé	763
Résoudre les problèmes liés à Amazon EC2 Auto Scaling	764
Résolution générale des problèmes liés à Amazon EC2 Auto Scaling	764
« CodeDeployRole ne vous autorise pas à effectuer des opérations dans le AWS service suivant : AmazonAutoScaling » erreur	765
Les instances d'un groupe Amazon EC2 Auto Scaling sont continuellement provisionnées et mises hors service avant qu'une révision ne puisse être déployée	766
La mise hors service ou le redémarrage d'une instance Amazon EC2 Auto Scaling peut entraîner l'échec des déploiements	767

Évitez d'associer plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling	768
Les instances EC2 d'un groupe Amazon EC2 Auto Scaling ne se lancent pas et reçoivent l'erreur « Heartbeat Timeout »	768
Des hooks de cycle de vie Amazon EC2 Auto Scaling mal adaptés peuvent entraîner l'arrêt ou l'échec des déploiements automatiques vers les groupes Amazon EC2 Auto Scaling	771
Erreur « Le déploiement a échoué car aucune instance n'a été trouvée pour votre groupe de déploiement »	773
Codes d'erreur	781
Rubriques en relation	786
Ressources	787
Guides de référence et ressources d'assistance	787
Exemples	787
Blogs	787
AWS kits et outils de développement logiciel	788
Historique du document	789
Mises à jour antérieures	809
AWS Glossaire	834
.....	dccccxxv

Qu'est-ce que c'est CodeDeploy ?

CodeDeploy est un service de déploiement qui automatise les déploiements d'applications sur des instances Amazon EC2, des instances sur site, des fonctions Lambda sans serveur ou des services Amazon ECS.

Vous pouvez déployer une grande variété de contenu d'application, y compris les éléments suivants :

- Code
- Fonctions sans serveur AWS Lambda
- Fichiers web et de configuration
- Executables (Fichiers exécutables)
- Forfaits
- Scripts
- Fichiers multimédias

CodeDeploy peut déployer du contenu d'application qui s'exécute sur un serveur et qui est stocké dans des compartiments Amazon S3, des GitHub référentiels ou des référentiels Bitbucket.

CodeDeploy peut également déployer une fonction Lambda sans serveur. Vous n'avez pas besoin d'apporter des modifications à votre code existant pour utiliser CodeDeploy.

CodeDeploy vous permet de :

- Publier de nouveaux éléments.
- Mettez à jour les versions des AWS Lambda fonctions.
- Éviter les temps d'arrêt lors du déploiement d'application.
- Gérer la complexité des mises à jour de vos applications, sans nombre des risques associés aux déploiements manuels sources d'erreurs.

Le service se dimensionne avec votre infrastructure de manière à ce que vous puissiez facilement déployer une ou des milliers d'instances.

CodeDeploy fonctionne avec différents systèmes pour la gestion de la configuration, le contrôle des sources, [l'intégration continue](#), [la livraison](#) continue et le déploiement continu. Pour plus d'informations, consultez la section [Intégrations de produits](#).

La CodeDeploy 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 aux ressources) 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 [Gestion des identités et des accès pour AWS CodeDeploy](#).

Rubriques

- [Avantages de AWS CodeDeploy](#)
- [Vue d'ensemble des plateformes CodeDeploy informatiques](#)
- [Vue d'ensemble des types de CodeDeploy déploiement](#)
- [Nous voulons entendre parler de vous](#)
- [CodeDeploy composants principaux](#)
- [CodeDeploy déploiements](#)
- [CodeDeploy fichiers de spécification de l'application \(AppSpec\)](#)

Avantages de AWS CodeDeploy

CodeDeploy offre les avantages suivants :

- Applications serveur, sans serveur et conteneurs. CodeDeploy vous permet de déployer à la fois des applications traditionnelles sur des serveurs et des applications qui déploient une version AWS Lambda fonctionnelle sans serveur ou une application Amazon ECS.
- Déploiements automatisés. CodeDeploy automatise entièrement le déploiement de vos applications dans vos environnements de développement, de test et de production. CodeDeploy s'adapte à votre infrastructure afin que vous puissiez le déployer sur une ou plusieurs milliers d'instances.
- Temps d'arrêt minimaux. Si votre application utilise la plate-forme de calcul EC2 sur site, CodeDeploy cela permet d'optimiser la disponibilité de votre application. Lors d'un déploiement sur place, CodeDeploy effectue une mise à jour continue sur les instances Amazon EC2. Vous pouvez spécifier le nombre d'instances à mettre hors ligne à la fois pour les mises à jour. Lors d'un blue/green déploiement, la dernière version de l'application est installée sur les instances de remplacement. Le trafic est réacheminé vers ces instances en fonction de ce que vous définissez : soit immédiatement ou dès que vous avez terminé de tester le nouvel environnement. Pour les

deux types de déploiement, CodeDeploy suit l'état de l'application selon les règles que vous configurez.

- Arrêter et restaurer. Vous pouvez arrêter et restaurer les déploiements automatiquement et manuellement en cas d'erreurs.
- Contrôle centralisé. Vous pouvez lancer et suivre l'état de vos déploiements via la CodeDeploy console ou le AWS CLI. Vous recevez un rapport qui indique à quel moment chaque révision d'application a été déployée et sur quelles instances Amazon EC2.
- Facile à adopter CodeDeploy est indépendant de la plate-forme et fonctionne avec n'importe quelle application. Vous pouvez facilement réutiliser votre code de configuration. CodeDeploy peut également s'intégrer à votre processus de publication logicielle ou à votre chaîne d'outils de livraison continue.
- Déploiements simultanés. Si plusieurs applications utilisent la plate-forme de calcul EC2/sur site, vous CodeDeploy pouvez les déployer simultanément sur le même ensemble d'instances.

Vue d'ensemble des plateformes CodeDeploy informatiques

CodeDeploy est capable de déployer des applications sur trois plateformes informatiques :

- EC2/sur site : décrit les instances de serveurs physiques qui peuvent être des instances cloud Amazon EC2, des serveurs sur site ou les deux. Les applications créées à l'aide de la plate-forme informatique EC2/on-premises peuvent être composées de fichiers exécutables, de fichiers de configuration, d'images, etc.

Les déploiements qui utilisent la plate-forme informatique EC2/sur site gèrent la manière dont le trafic est dirigé vers les instances en utilisant un type de déploiement sur place ou bleu/vert. Pour de plus amples informations, veuillez consulter [Vue d'ensemble des types de CodeDeploy déploiement](#).

- AWS Lambda : utilisé pour déployer des applications qui consistent en une version mise à jour d'une fonction Lambda. AWS Lambda gère la fonction Lambda dans un environnement de calcul sans serveur composé d'une structure de calcul à haute disponibilité. Toute l'administration des ressources informatiques est effectuée par AWS Lambda. Pour plus d'informations, consultez la section [Informatique et applications sans serveur](#). Pour plus d'informations sur AWS Lambda les fonctions Lambda, consultez. [AWS Lambda](#)

Vous pouvez gérer la manière dont le trafic est transféré vers les versions mises à jour des fonctions Lambda lors d'un déploiement en choisissant une option Canary, une configuration linéaire ou all-at-once une configuration.

- Amazon ECS : utilisé pour déployer une application conteneurisée Amazon ECS sous forme d'ensemble de tâches. CodeDeploy effectue un blue/green déploiement en installant une version mise à jour de l'application en tant que nouvel ensemble de tâches de remplacement. CodeDeploy redirige le trafic de production de l'ensemble de tâches d'application d'origine vers le jeu de tâches de remplacement. L'ensemble de tâches d'origine est arrêté après un déploiement réussi. Pour plus d'informations sur Amazon ECS, consultez [Amazon Elastic Container Service](#).

Vous pouvez gérer la manière dont le trafic est transféré vers l'ensemble de tâches mis à jour au cours d'un déploiement en choisissant une configuration Canary, Linear ou une all-at-once configuration.

 Note

Les blue/green déploiements Amazon ECS sont pris en charge en utilisant à la fois CodeDeploy et CloudFormation. Les détails de ces déploiements sont décrits dans les sections suivantes.

Le tableau suivant décrit comment CodeDeploy les composants sont utilisés avec chaque plate-forme de calcul. Pour en savoir plus, consultez :

- [Utilisation de groupes de déploiement dans CodeDeploy](#)
- [Travailler avec des déploiements dans CodeDeploy](#)
- [Utilisation des configurations de déploiement dans CodeDeploy](#)
- [Travailler avec les révisions d'applications pour CodeDeploy](#)
- [Utilisation d'applications dans CodeDeploy](#)

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
Groupe de déploiement	Déploie une révision sur un ensemble d'instances.	Déploie une nouvelle version d'une fonction Lambda sans serveur	Spécifie le service Amazon

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
		sur une infrastructure informatique à haute disponibilité.	ECS avec l'application conteneurisée à déployer sous forme d'ensemble de tâches, un écouteur de production et de test facultatif utilisé pour acheminer le trafic vers l'application déployée, le moment où il convient de rediriger le trafic et de mettre fin à l'ensemble de tâches initial de l'application déployée, ainsi

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
			que les paramètre s de déclenche ment, d'alarme et de restaurat ion facultati fs.

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
Déploiement	Déploie une nouvelle révision composée d'une application et d'un AppSpec fichier. AppSpec Spécifie comment déployer l'application sur les instances d'un groupe de déploiement.	Déplace le trafic de production d'une version d'une fonction Lambda vers une nouvelle version de la même fonction. Le AppSpec fichier indique la version de la fonction Lambda à déployer.	Déploie une version mise à jour d'une application conteneur isée Amazon ECS en tant que nouvel ensemble de tâches de remplacement. CodeDeploy redirige le trafic de production de l'ensemble de tâches avec la version d'origine vers le nouvel ensemble de tâches de remplacement avec

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
			la version mise à jour. Lorsque le déploiement est terminé, l'ensemble de tâches d'origine est supprimé.
Configuration de déploiement	Les paramètres déterminant la vitesse de déploiement et le nombre minimum d'instances devant être sains à tout moment d'un déploiement.	Paramètres qui déterminent la manière dont le trafic est transféré vers les versions mises à jour de la fonction Lambda.	Paramètres qui déterminent la manière dont le trafic est transféré vers l'ensemble de tâches Amazon ECS mis à jour.

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
Revision	Combinaison d'un AppSpec fichier et de fichiers d'application, tels que des exécutables, des fichiers de configuration, etc.	AppSpec Fichier qui spécifie la fonction Lambda à déployer et les fonctions Lambda qui peuvent exécuter des tests de validation lors des accroches d'événements du cycle de vie du déploiement.	Un AppSpec fichier qui spécifie : • Définition de tâche Amazon ECS pour le service Amazon ECS avec l'application conteneurisée à déployer. • Le conteneur où votre application mise à jour est déployée. • Un port pour le conteneur où le trafic de production est

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
			réacheminé. Paramètres de configuration réseau facultatifs et fonctions Lambda permettant d'exécuter des tests de validation lors des accrochages d'événements du cycle de vie du déploiement.

CodeDeploy composant	EC2/Solutions sur site	AWS Lambda	Amazon ECS
Application	Une collection de groupes de déploiement et de révisions. Une application EC2/sur site utilise la plate-forme de calcul EC2/sur site.	Une collection de groupes de déploiement et de révisions. Une application utilisée pour un AWS Lambda déploiement utilise la plate-forme de calcul AWS Lambda sans serveur.	Une collection de groupes de déploiement et de révisions. Une application utilisée pour un déploiement Amazon ECS utilise la plateforme de calcul Amazon ECS.

Vue d'ensemble des types de CodeDeploy déploiement

CodeDeploy propose deux options de type de déploiement :

- **Déploiement sur place** : l'application de chaque instance du groupe de déploiement est arrêtée, la dernière révision de l'application est installée et la nouvelle version de l'application est démarrée et validée. Vous pouvez utiliser un équilibreur de charge afin que chaque instance soit désenregistrée pendant son déploiement, puis remise en service une fois le déploiement terminé. Seuls les déploiements qui utilisent la plate-forme informatique EC2/sur site peuvent utiliser des déploiements sur place. Pour plus d'informations sur les déploiements sur place, consultez [Vue d'ensemble d'un déploiement sur place](#).

 Note

AWS Les déploiements Lambda et Amazon ECS ne peuvent pas utiliser un type de déploiement sur place.

- Déploiement bleu/vert : le comportement de votre déploiement dépend de la plateforme informatique que vous utilisez :
 - Blue/green on an EC2/On-Plateforme de calcul sur site : les instances d'un groupe de déploiement (l'environnement d'origine) sont remplacées par un ensemble d'instances différent (l'environnement de remplacement) en suivant les étapes suivantes :
 - Les instances sont provisionnées pour l'environnement de remplacement.
 - La dernière version de l'application est installée sur les instances de remplacement.
 - Un temps d'attente facultatif se produit pour des activités telles que les tests d'applications et la vérification du système.
 - Les instances de l'environnement de remplacement sont enregistrées auprès d'un ou de plusieurs équilibreurs de charge Elastic Load Balancing, ce qui entraîne le reroutage du trafic vers celles-ci. Les instances de l'environnement d'origine sont annulées et peuvent être résiliées ou maintenues en cours d'exécution pour d'autres utilisations.

 Note

Si vous utilisez une plate-forme de calcul EC2/sur site, sachez que les blue/green déploiements fonctionnent uniquement avec les instances Amazon EC2.

- Bleu/vert sur une plateforme de calcul AWS Lambda Amazon ECS ou Amazon ECS : le trafic est décalé par incréments en fonction d'une configuration canarienne, linéaire ou all-at-once de déploiement.
- Déploiements bleu/vert via CloudFormation : le trafic est transféré de vos ressources actuelles vers vos ressources mises à jour dans le cadre d'une CloudFormation mise à jour de la pile. Actuellement, seuls les blue/green déploiements ECS sont pris en charge.

Pour plus d'informations sur les blue/green déploiements, consultez [Vue d'ensemble d'un blue/green déploiement](#).

Note

À l'aide de l' CodeDeploy agent, vous pouvez effectuer un déploiement sur une instance à laquelle vous êtes connecté sans avoir besoin d'une application, d'un groupe de déploiement ou même d'un AWS compte. Pour plus d'informations, consultez [Utiliser l' CodeDeploy agent pour valider un package de déploiement sur une machine locale](#).

Rubriques

- [Vue d'ensemble d'un déploiement sur place](#)
- [Vue d'ensemble d'un blue/green déploiement](#)

Vue d'ensemble d'un déploiement sur place

Note

AWS Les déploiements Lambda et Amazon ECS ne peuvent pas utiliser un type de déploiement sur place.

Voici comment fonctionne un déploiement sur place :

1. Tout d'abord, vous créez du contenu déployable sur votre machine de développement locale ou dans un environnement similaire, puis vous ajoutez un fichier de spécification d'application (AppSpec fichier). Le AppSpec fichier est unique à CodeDeploy. Il définit les actions de déploiement que vous CodeDeploy souhaitez exécuter. Vous regroupez votre contenu déployable et le AppSpec fichier dans un fichier d'archive, puis vous le chargez dans un compartiment ou un GitHub référentiel Amazon S3. Ce fichier d'archive est appelé une révision d'application (ou simplement une révision).
2. Ensuite, vous fournissez CodeDeploy des informations sur votre déploiement, telles que le compartiment ou le GitHub référentiel Amazon S3 d'où vous souhaitez extraire la révision et vers quel ensemble d'instances Amazon EC2 déployer son contenu. CodeDeploy appelle un ensemble d'instances Amazon EC2 un groupe de déploiement. Un groupe de déploiement contient des instances Amazon EC2 étiquetées individuellement, des instances Amazon EC2 appartenant à des groupes Amazon EC2 Auto Scaling, ou les deux.

Chaque fois que vous chargez avec succès une nouvelle révision d'application que vous souhaitez déployer dans le groupe de déploiement, ce paquet est défini comme la révision cible pour le groupe de déploiement. En d'autres termes, il s'agit de la révision d'application actuellement ciblée pour le déploiement. C'est également la révision utilisée pour les déploiements automatiques.

3. Ensuite, l' agent CodeDeploy de chaque instance interroge CodeDeploy pour déterminer ce qu'il faut extraire du compartiment ou du GitHub référentiel Amazon S3 spécifié et à quel moment.
4. Enfin, l' agent CodeDeploy de chaque instance extrait la révision cible du compartiment ou du GitHub référentiel Amazon S3 et, à l'aide des instructions du AppSpec fichier, déploie le contenu sur l'instance.

CodeDeploy conserve un enregistrement de vos déploiements afin que vous puissiez obtenir l'état du déploiement, les paramètres de configuration du déploiement, l'état de santé de l'instance, etc.

Vue d'ensemble d'un blue/green déploiement

Un blue/green déploiement permet de mettre à jour vos applications tout en minimisant les interruptions causées par les modifications apportées à une nouvelle version de l'application. CodeDeploy approvisionne la nouvelle version de votre application en même temps que l'ancienne version avant de rediriger votre trafic de production.

- AWS Lambda : le trafic est transféré d'une version d'une fonction Lambda à une nouvelle version de la même fonction Lambda.
- Amazon ECS : le trafic est transféré d'un ensemble de tâches de votre service Amazon ECS à un ensemble de tâches de remplacement mis à jour dans le même service Amazon ECS.
- EC2/sur site : le trafic est transféré d'un ensemble d'instances de l'environnement d'origine vers un ensemble d'instances de remplacement.

Tous les déploiements AWS Lambda et Amazon ECS offrent de nombreux avantages par rapport à un blue/green. An EC2/On-Premises deployment can be in-place or blue/green. A blue/green déploiement sur place :

- Vous pouvez installer et tester une application dans le nouvel environnement de remplacement et la déployer en production simplement en réacheminant le trafic.
- Si vous utilisez la plate-forme informatique EC2/on-premises, le retour à la version la plus récente d'une application est plus rapide et plus fiable. En effet, le trafic peut être réacheminé vers les

instances d'origine tant qu'elles n'ont pas été mises hors service. Avec un déploiement sur place, les versions doivent être restaurées en redéployant la version précédente de l'application.

- Si vous utilisez la plate-forme de calcul EC2/on-premises, les nouvelles instances sont provisionnées pour un blue/green déploiement et reflètent la plupart des configurations de serveur. up-to-date Cela vous aide à éviter le genre de problèmes se produisant parfois sur des instances à longue opération.
- Si vous utilisez la plate-forme de calcul AWS Lambda, vous contrôlez la manière dont le trafic est transféré de la version d'origine de votre fonction AWS Lambda à votre nouvelle version de fonction AWS Lambda.
- Si vous utilisez la plate-forme de calcul Amazon ECS, vous contrôlez la manière dont le trafic est transféré de votre ensemble de tâches d'origine vers votre nouvel ensemble de tâches.

Un blue/green déploiement avec CloudFormation peut utiliser l'une des méthodes suivantes :

- CloudFormation modèles pour les déploiements : lorsque vous configurez des déploiements à l'aide de CloudFormation modèles, vos déploiements sont déclenchés par des mises à jour. CloudFormation Lorsque vous modifiez une ressource et que vous téléchargez une modification de modèle, une mise à jour de la pile CloudFormation initie le nouveau déploiement. Pour obtenir la liste des ressources que vous pouvez utiliser dans les CloudFormation modèles, consultez [CloudFormation modèles de CodeDeploy référence](#).
- Déploiements bleu/vert via CloudFormation : vous pouvez les utiliser CloudFormation pour gérer vos blue/green déploiements par le biais de mises à jour de la pile. Vous définissez vos ressources bleues et vertes, en plus de spécifier les paramètres de routage et de stabilisation du trafic, dans le modèle de pile. Ensuite, si vous mettez à jour les ressources sélectionnées lors d'une mise à jour de la pile, que vous CloudFormation générez toutes les ressources vertes nécessaires, que vous déplacez le trafic en fonction des paramètres de routage du trafic spécifiés et que vous supprimez les ressources bleues. Pour plus d'informations, consultez [Automatiser les blue/green déploiements d'Amazon ECS en les CodeDeploy utilisant CloudFormation](#) dans le Guide de l'AWS CloudFormation utilisateur.

 Note

Pris en charge uniquement pour les blue/green déploiements Amazon ECS.

La façon dont vous configurez un blue/green déploiement dépend de la plate-forme informatique utilisée par votre déploiement.

Déploiement bleu/vert sur une plateforme AWS Lambda informatique ou Amazon ECS

Si vous utilisez la plateforme de calcul Amazon ECS AWS Lambda ou Amazon ECS, vous devez indiquer comment le trafic est transféré de la AWS Lambda fonction ou de l'ensemble de tâches Amazon ECS d'origine vers la nouvelle fonction ou le nouvel ensemble de tâches. Pour indiquer comment le trafic est transféré, vous devez spécifier l'une des configurations de déploiement suivantes :

- canari
- linéaire
- all-at-once

Pour plus d'informations sur la manière dont le trafic est déplacé dans une configuration Canary, linéaire ou de all-at-once déploiement, consultez [Configuration de déploiement](#).

Pour plus de détails sur la configuration de déploiement Lambda, consultez. [Configurations de déploiement sur une plateforme AWS Lambda informatique](#)

Pour plus de détails sur la configuration de déploiement d'Amazon ECS, consultez [Configurations de déploiement sur une plateforme de calcul Amazon ECS](#).

Blue/Green deployment on an EC2/onplate-forme informatique sur site

Note

Vous devez utiliser des instances Amazon EC2 pour les blue/green déploiements sur la plate-forme de calcul EC2/sur site. Les instances locales ne sont pas prises en charge pour le type de blue/green déploiement.

Si vous utilisez la plate-forme de calcul EC2/sur site, les règles suivantes s'appliquent :

Vous devez disposer d'une ou de plusieurs instances Amazon EC2 avec des balises Amazon EC2 d'identification ou d'un groupe Amazon EC2 Auto Scaling. Les instances doivent répondre à ces exigences supplémentaires :

- Le profil d'instance IAM approprié doit être attaché à chaque instance Amazon EC2.
- L' agent CodeDeploy doit être installé et exécuté sur chaque instance.

 Note

En général, une révision d'application est également exécutée sur les instances de votre environnement d'origine, mais cela n'est pas obligatoire pour un blue/green déploiement.

Lorsque vous créez un groupe de déploiement utilisé dans les blue/green déploiements, vous pouvez choisir la manière dont votre environnement de remplacement est spécifié :

Copiez un groupe Amazon EC2 Auto Scaling existant : pendant blue/green le déploiement CodeDeploy , crée les instances pour votre environnement de remplacement pendant le déploiement. Avec cette option, CodeDeploy utilise le groupe Amazon EC2 Auto Scaling que vous spécifiez comme modèle pour l'environnement de remplacement, y compris le même nombre d'instances en cours d'exécution et de nombreuses autres options de configuration.

Choisissez les instances manuellement : vous pouvez spécifier les instances à considérer comme des instances de remplacement à l'aide des balises d'instance Amazon EC2, des noms de groupes Amazon EC2 Auto Scaling, ou des deux. Si vous sélectionnez cette option, vous n'avez pas besoin de spécifier les instances de l'environnement de remplacement jusqu'à la création d'un déploiement.

Voici comment cela fonctionne :

1. Vous possédez déjà des instances ou un groupe Amazon EC2 Auto Scaling qui constitue votre environnement d'origine. La première fois que vous exécutez un blue/green déploiement, vous utilisez généralement des instances déjà utilisées dans le cadre d'un déploiement sur place.
2. Dans une CodeDeploy application existante, vous créez un groupe de blue/green déploiement dans lequel, outre les options requises pour un déploiement sur place, vous spécifiez les éléments suivants :
 - Le ou les équilibres de charge qui acheminent le trafic de votre environnement d'origine vers votre environnement de remplacement pendant le processus de blue/green déploiement.
 - Réacheminement immédiat du trafic vers l'environnement de remplacement ou attente pour le réacheminer manuellement.
 - Débit auquel le trafic est acheminé vers les instances de remplacement.
 - Si les instances remplacées sont mises hors service ou si elles continuent de s'exécuter.

3. Vous créez un déploiement pour ce groupe de déploiement au cours duquel se produisent les événements suivants :
 - a. Si vous avez choisi de copier un groupe Amazon EC2 Auto Scaling, des instances sont mises en service pour votre environnement de remplacement.
 - b. La révision d'application que vous spécifiez pour le déploiement est installée sur les instances de remplacement.
 - c. Si vous avez spécifié un temps d'attente dans les paramètres de groupe de déploiement, le déploiement est interrompu. C'est le moment où vous pouvez exécuter les tests et vérifications dans votre environnement de remplacement. Si vous ne réacheminez pas manuellement le trafic avant la fin du délai d'attente, le déploiement est arrêté.
 - d. Les instances de l'environnement de remplacement sont enregistrées auprès d'un équilibreur de charge Elastic Load Balancing et le trafic commence à y être acheminé.
 - e. L'enregistrement des instances de l'environnement d'origine est annulé et elles sont traitées selon vos spécifications dans le groupe de déploiement. Soit elles sont mises hors service, soit elles continuent de s'exécuter.

Déploiement bleu/vert via CloudFormation

Vous pouvez gérer les déploiements CodeDeploy bleu/vert en modélisant vos ressources à l'aide d'un modèle. CloudFormation

Lorsque vous modélisez vos blue/green ressources à l'aide d'un CloudFormation modèle, vous créez une mise à jour de la pile CloudFormation qui met à jour votre ensemble de tâches. Le trafic de production passe de l'ensemble de tâches d'origine de votre service à un ensemble de tâches de remplacement soit en une seule fois, avec des déploiements linéaires et des temps d'attente, soit avec des déploiements de type Canary. La mise à jour de la pile lance un déploiement dans CodeDeploy. Vous pouvez consulter l'état et l'historique du déploiement dans le modèle CodeDeploy, mais vous ne devez pas créer ni gérer de CodeDeploy ressources en dehors du CloudFormation modèle.

Note

Pour les blue/green déploiements via CloudFormation, vous ne créez pas d' CodeDeploy application ou de groupe de déploiement.

Cette méthode prend uniquement en charge les blue/green déploiements Amazon ECS. Pour plus d'informations sur les blue/green déploiements via CloudFormation, consultez [Créez un blue/green déploiement Amazon ECS via CloudFormation](#).

Nous voulons entendre parler de vous

Nous apprécions vos commentaires. Pour nous contacter, rendez-vous [CodeDeploy sur le forum](#).

Rubriques

- [Primary Components](#)
- [Deployments](#)
- [Application Specification Files](#)

CodeDeploy composants principaux

Avant de commencer à utiliser le service, vous devez vous familiariser avec les principaux composants du processus de CodeDeploy déploiement.

Rubriques

- [Application](#)
- [Plateforme de calcul](#)
- [Configuration de déploiement](#)
- [Groupe de déploiement](#)
- [Type de déploiement](#)
- [Profil d'instance IAM](#)
- [Revision](#)
- [Rôle de service](#)
- [Révision cible](#)
- [Autres composants](#)

Application

Une application est un nom qui identifie de manière unique l'application que vous souhaitez déployer. CodeDeploy utilise ce nom, qui fonctionne comme un conteneur, pour garantir que la combinaison

correcte de révision, de configuration de déploiement et de groupe de déploiement est référencée lors d'un déploiement.

Plateforme de calcul

Une plateforme de calcul est une plateforme sur laquelle est CodeDeploy déployée une application. Il existe trois plateformes de calcul :

- EC2/On-Premises : décrit les instances de serveurs physiques qui peuvent être des instances EC2 cloud Amazon, des serveurs sur site ou les deux. Les applications créées à l'aide de la plateforme EC2 de calcul /On-Premises peuvent être composées de fichiers exécutables, de fichiers de configuration, d'images, etc.

Les déploiements qui utilisent la plate-forme de calcul EC2 /On-Premises gèrent la manière dont le trafic est dirigé vers les instances en utilisant un type de déploiement sur place ou bleu/vert.

Pour de plus amples informations, veuillez consulter [Vue d'ensemble des types de CodeDeploy déploiement](#).

- AWS Lambda : utilisé pour déployer des applications qui consistent en une version mise à jour d'une fonction Lambda. AWS Lambda gère la fonction Lambda dans un environnement de calcul sans serveur composé d'une structure de calcul à haute disponibilité. Toute l'administration des ressources informatiques est effectuée par AWS Lambda. Pour plus d'informations, consultez la section [Informatique et applications sans serveur](#). Pour plus d'informations sur AWS Lambda les fonctions Lambda, consultez. [AWS Lambda](#)

Vous pouvez gérer la manière dont le trafic est transféré vers les versions mises à jour des fonctions Lambda lors d'un déploiement en choisissant une option Canary, une configuration linéaire ou all-at-once une configuration.

- Amazon ECS : utilisé pour déployer une application conteneurisée Amazon ECS sous forme d'ensemble de tâches. CodeDeploy effectue un blue/green déploiement en installant une version mise à jour de l'application en tant que nouvel ensemble de tâches de remplacement. CodeDeploy redirige le trafic de production de l'ensemble de tâches d'application d'origine vers le jeu de tâches de remplacement. L'ensemble de tâches d'origine est arrêté après un déploiement réussi. Pour plus d'informations sur Amazon ECS, consultez [Amazon Elastic Container Service](#).

Vous pouvez gérer la manière dont le trafic est transféré vers l'ensemble de tâches mis à jour au cours d'un déploiement en choisissant une configuration Canary, Linear ou une all-at-once configuration.

 Note

Les blue/green déploiements Amazon ECS sont pris en charge à la fois par CodeDeploy et CloudFormation. Les détails de ces déploiements sont décrits dans les sections suivantes.

Configuration de déploiement

Une configuration de déploiement est un ensemble de règles de déploiement et de conditions de réussite et d'échec du déploiement utilisées CodeDeploy lors d'un déploiement. Si votre déploiement utilise la plate-forme de calcul EC2 /On-Premises, vous pouvez spécifier le nombre minimum d'instances saines pour le déploiement. Si votre déploiement utilise la plateforme de calcul Amazon ECS AWS Lambda ou la plateforme de calcul Amazon ECS, vous pouvez spécifier la manière dont le trafic est acheminé vers votre fonction Lambda ou votre ensemble de tâches ECS mis à jour.

Pour plus d'informations sur la spécification du nombre minimum d'hôtes sains pour un déploiement utilisant la plate-forme de calcul EC2 /On-Premises, consultez. [À propos du nombre minimum d'instances saines](#)

Les configurations de déploiement suivantes spécifient la manière dont le trafic est acheminé lors d'un déploiement utilisant la plate-forme de calcul Lambda ou ECS :

- **Canary** : le trafic est déplacé en deux incréments. Vous pouvez choisir parmi les options Canary prédéfinies qui spécifient le pourcentage de trafic transféré vers votre fonction Lambda ou votre ensemble de tâches ECS mis à jour dans le premier incrément et l'intervalle, en minutes, avant que le trafic restant ne soit transféré dans le second incrément.
- **Linéaire** : le trafic est déplacé en incréments égaux, avec un nombre de minutes égal entre chaque incrément. Vous pouvez choisir parmi les options linéaires prédéfinies qui définissent le pourcentage de trafic déplacé pour chaque incrément et le nombre de minutes entre chaque incrément.
- **R II-at-once** : Tout le trafic est transféré de la fonction Lambda ou de l'ensemble de tâches ECS d'origine vers la fonction ou l'ensemble de tâches mis à jour en une seule fois.

Groupe de déploiement

Un groupe de déploiement est un ensemble d'instances individuelles. Un groupe de déploiement contient des instances étiquetées individuellement, des EC2 instances Amazon dans des groupes

Amazon EC2 Auto Scaling, ou les deux. Pour plus d'informations sur les balises d' EC2 instance Amazon, consultez la section [Utilisation des balises à l'aide de la console](#). Pour en savoir plus sur les instances sur site, consultez la rubrique [Working with On-Premises Instances](#). Pour plus d'informations sur Amazon EC2 Auto Scaling, consultez [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).

Type de déploiement

Un type de déploiement est une méthode utilisée pour rendre la dernière révision de l'application disponible sur les instances d'un groupe de déploiement. Il existe deux types de déploiement :

- **Déploiement sur place** : l'application de chaque instance du groupe de déploiement est arrêtée, la dernière révision de l'application est installée et la nouvelle version de l'application est démarrée et validée. Vous pouvez utiliser un équilibreur de charge afin que chaque instance soit désenregistrée pendant son déploiement, puis remise en service une fois le déploiement terminé. Seuls les déploiements qui utilisent la plate-forme de calcul EC2 /On-Premises peuvent utiliser des déploiements sur place. Pour plus d'informations sur les déploiements sur place, consultez [Vue d'ensemble d'un déploiement sur place](#).
- **Déploiement bleu/vert** : le comportement de votre déploiement dépend de la plateforme informatique que vous utilisez :
 - **Blue/green on an EC2/On-Plateforme de calcul sur site** : les instances d'un groupe de déploiement (l'environnement d'origine) sont remplacées par un ensemble d'instances différent (l'environnement de remplacement) en suivant les étapes suivantes :
 - Les instances sont provisionnées pour l'environnement de remplacement.
 - La dernière version de l'application est installée sur les instances de remplacement.
 - Un temps d'attente facultatif se produit pour des activités telles que les tests d'applications et la vérification du système.
 - Les instances de l'environnement de remplacement sont enregistrées auprès d'un ou de plusieurs équilibreurs de charge Elastic Load Balancing, ce qui entraîne le reroutage du trafic vers celles-ci. Les instances de l'environnement d'origine sont désenregistrées et peuvent être résiliées ou maintenues en cours d'exécution pour d'autres utilisations.

Note

Si vous utilisez une plateforme de calcul EC2 /On-Premises, sachez que les blue/green déploiements fonctionnent uniquement avec les instances Amazon EC2 .

- Bleu/vert sur une plateforme de calcul AWS Lambda Amazon ECS ou Amazon ECS : le trafic est décalé par incréments en fonction d'une configuration canarienne, linéaire ou all-at-once de déploiement.
- Déploiements bleu/vert via CloudFormation : le trafic est transféré de vos ressources actuelles vers vos ressources mises à jour dans le cadre d'une CloudFormation mise à jour de la pile. Actuellement, seuls les blue/green déploiements ECS sont pris en charge.

Pour plus d'informations sur les blue/green déploiements, consultez [Vue d'ensemble d'un blue/green déploiement](#).

Note

Les blue/green déploiements Amazon ECS sont pris en charge en utilisant à la fois CodeDeploy et CloudFormation. Les détails de ces déploiements sont décrits dans les sections suivantes.

Profil d'instance IAM

Un profil d'instance IAM est un rôle IAM que vous associez à vos instances Amazon EC2. Ce profil inclut les autorisations requises pour accéder aux compartiments ou aux référentiels Amazon S3 dans lesquels les applications sont stockées. Pour de plus amples informations, veuillez consulter [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).

Revision

Une révision est une version de votre application. Une révision de déploiement AWS Lambda est un fichier au format YAML ou JSON qui spécifie des informations sur la fonction Lambda à déployer. Une révision de déploiement EC2 /On-Premises est un fichier d'archive qui contient le contenu source (code source, pages Web, fichiers exécutables et scripts de déploiement) et un fichier de spécification d'application (AppSpec fichier). AWS Les révisions Lambda peuvent être stockées dans des compartiments Amazon S3. EC2 Les révisions /On-Premises sont stockées dans des compartiments ou des référentiels Amazon S3. GitHub Pour Amazon S3, une révision est identifiée de manière unique par sa clé d'objet Amazon S3 et sa version ETag, ou les deux. En GitHub effet, une révision est identifiée de manière unique par son ID de validation.

Rôle de service

Un rôle de service est un rôle IAM qui accorde des autorisations à un AWS service afin qu'il puisse accéder aux AWS ressources. Les politiques que vous attachez au rôle de service déterminent les AWS ressources auxquelles le service peut accéder et les actions qu'il peut effectuer avec ces ressources. En CodeDeploy effet, un rôle de service est utilisé pour les opérations suivantes :

- Pour lire les balises appliquées aux instances ou les noms de groupes Amazon EC2 Auto Scaling associés aux instances. Cela permet d' CodeDeploy identifier les instances sur lesquelles il peut déployer des applications.
- Pour effectuer des opérations sur des instances, des groupes Amazon EC2 Auto Scaling et des équilibresurs de charge Elastic Load Balancing.
- Publier des informations dans les rubriques Amazon SNS afin que des notifications puissent être envoyées lorsque des événements de déploiement ou d'instance spécifiés se produisent.
- Pour récupérer des informations sur les CloudWatch alarmes afin de configurer la surveillance des alarmes pour les déploiements.

Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

Révision cible

Une révision cible est la version la plus récente de la révision de l'application que vous avez téléchargée dans votre référentiel et que vous souhaitez déployer sur les instances d'un groupe de déploiement. En d'autres termes, il s'agit de la révision d'application actuellement ciblée pour le déploiement. C'est également la révision utilisée pour les déploiements automatiques.

Autres composants

Pour plus d'informations sur les autres composants du CodeDeploy flux de travail, consultez les rubriques suivantes :

- [Choisissez un type de CodeDeploy référentiel](#)
- [Deployments](#)
- [Application Specification Files](#)
- [Instance Health](#)

- [Travailler avec l' CodeDeploy agent](#)
- [Working with On-Premises Instances](#)

CodeDeploy déploiements

Cette rubrique fournit des informations sur les composants et le flux de travail des déploiements dans CodeDeploy. Le processus de déploiement varie en fonction de la plateforme de calcul ou de la méthode de déploiement (Lambda, Amazon ECS, EC2/on-premises ou via AWS CloudFormation) que vous utilisez pour vos déploiements.

Rubriques

- [Déploiements sur une plateforme de calcul AWS Lambda](#)
- [Déploiements sur une plateforme de calcul Amazon ECS](#)
- [Déploiements sur une plateforme informatique EC2/sur site](#)

Déploiements sur une plateforme de calcul AWS Lambda

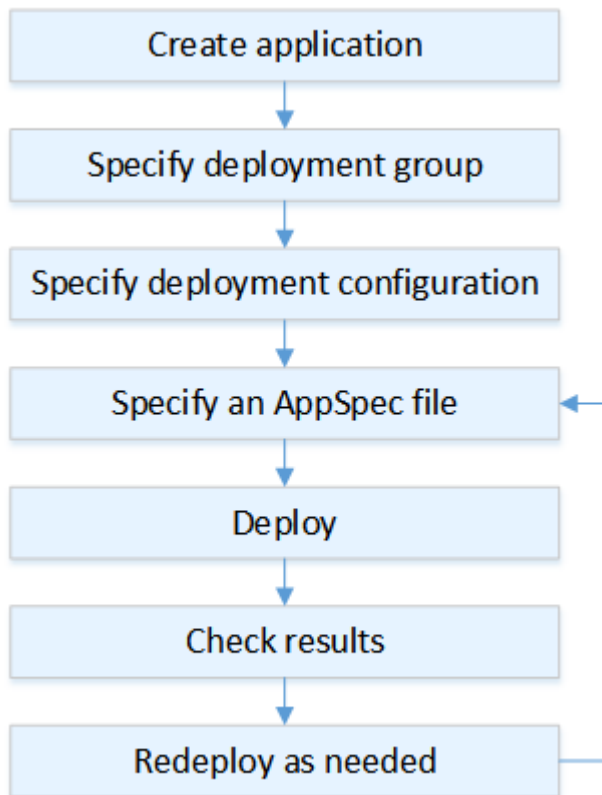
Cette rubrique fournit des informations sur les composants et le flux de travail des CodeDeploy déploiements qui utilisent la plate-forme de calcul AWS Lambda.

Rubriques

- [Flux de travail de déploiement sur une plateforme AWS Lambda informatique](#)
- [Téléchargement de la version révisée de votre application](#)
- [Création de vos groupes d'applications et de déploiement](#)
- [Déploiement de la révision de votre application](#)
- [Mettre à jour votre application](#)
- [Déploiements interrompus ou échoués](#)
- [Redéploiements et annulations de déploiements](#)

Flux de travail de déploiement sur une plateforme AWS Lambda informatique

Le schéma suivant illustre les principales étapes du déploiement de fonctions AWS Lambda nouvelles et mises à jour.



Les étapes sont les suivantes :

1. Créez une application et attribuez-lui un nom qui identifie de façon unique les révisions d'application que vous souhaitez déployer. Pour déployer des fonctions Lambda, choisissez la plateforme de calcul AWS Lambda lorsque vous créez votre application. CodeDeploy utilise ce nom lors d'un déploiement pour s'assurer qu'il fait référence aux composants de déploiement appropriés, tels que le groupe de déploiement, la configuration du déploiement et la révision de l'application. Pour de plus amples informations, veuillez consulter [Créez une application avec CodeDeploy](#).
2. Configurez un groupe de déploiement en spécifiant votre nom de groupe de déploiement.
3. Choisissez une configuration de déploiement pour spécifier la manière dont le trafic est transféré de votre version de AWS Lambda fonction d'origine à votre nouvelle version de fonction Lambda. Pour de plus amples informations, veuillez consulter [View Deployment Configuration Details](#).
4. Chargement d'un fichier de spécification d'application (AppSpec fichier) sur Amazon S3. Le AppSpec fichier indique la version de la fonction Lambda et les fonctions Lambda utilisées pour valider votre déploiement. Si vous ne souhaitez pas créer de AppSpec fichier, vous pouvez spécifier une version de fonction Lambda et des fonctions de validation du déploiement Lambda

directement dans la console à l'aide de YAML ou JSON. Pour de plus amples informations, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

5. Déployez la révision de votre application dans le groupe de déploiement. AWS CodeDeploy déploie la révision de la fonction Lambda que vous avez spécifiée. Le trafic est transféré vers votre version de fonction Lambda à l'aide du AppSpec fichier de déploiement que vous avez choisi lors de la création de votre application. Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).
6. Vérifiez les résultats du déploiement. Pour de plus amples informations, veuillez consulter [Surveillance des déploiements dans CodeDeploy](#).

Téléchargement de la version révisée de votre application

Placez un AppSpec fichier dans Amazon S3 ou saisissez-le directement dans la console ou AWS CLI. Pour de plus amples informations, veuillez consulter [Application Specification Files](#).

Création de vos groupes d'applications et de déploiement

Un groupe CodeDeploy de déploiement sur une plate-forme de calcul AWS Lambda identifie un ensemble d'un ou de plusieurs AppSpec fichiers. Chaque AppSpec fichier peut déployer une version de fonction Lambda. Un groupe de déploiement définit également un ensemble d'options de configuration pour les déploiements futurs, telles des alarmes et des configurations de rétrogradation.

Déploiement de la révision de votre application

Vous êtes maintenant prêt à déployer la révision de fonction spécifiée dans le AppSpec fichier vers le groupe de déploiement. Vous pouvez utiliser la CodeDeploy console ou la commande [create-deployment](#). Vous pouvez spécifier certains paramètres pour contrôler votre déploiement, y compris la révision, le groupe de déploiement et la configuration de déploiement.

Mettre à jour votre application

Vous pouvez mettre à jour votre application, puis utiliser la CodeDeploy console ou appeler la commande [create-deployment](#) pour envoyer une révision.

Déploiements interrompus ou échoués

Vous pouvez utiliser la CodeDeploy console ou la commande [stop-deployment](#) pour arrêter un déploiement. Lorsque vous essayez d'arrêter le déploiement, trois choses peuvent se produire :

- Le déploiement s'arrête et l'opération renvoie un statut de réussite. Dans ce cas, aucun événement de cycle de vie de déploiement supplémentaire ne se déroule sur le groupe de déploiement pour le déploiement arrêté.
- Le déploiement ne s'arrête pas immédiatement et l'opération renvoie un statut d'attente. Dans ce cas, certains événements de cycle de vie de déploiement peuvent encore se dérouler sur le groupe de déploiement. Une fois l'opération en attente terminée, les appels suivants visant à arrêter le déploiement renvoient un statut de réussite.
- Le déploiement ne peut pas s'arrêter et l'opération renvoie une erreur. Pour plus d'informations, consultez [ErrorInformation](#) la section [Erreurs courantes](#) dans la référence AWS CodeDeploy d'API.

Comme les déploiements arrêtés, les déploiements échoués peuvent se traduire par des événements de cycle de vie de déploiement ayant déjà été exécutés. Pour découvrir les raisons d'un déploiement échoué, vous pouvez utiliser la console CodeDeploy ou analyser les données du fichier journal à partir du déploiement ayant échoué. Pour plus d'informations, consultez [Révision de l'application et nettoyage des fichiers journaux](#) et [Afficher les données du journal pour les déploiements CodeDeploy EC2/sur site](#).

Redéploiements et annulations de déploiements

CodeDeploy implémente les annulations en redéployant, en tant que nouveau déploiement, une révision précédemment déployée.

Vous pouvez configurer un groupe de déploiement pour restaurer automatiquement les déploiements lorsque certaines conditions sont remplies, y compris lorsqu'un déploiement échoue, ou si un seuil de surveillance d'alarme est atteint. Vous pouvez également remplacer les paramètres de restauration spécifiés pour un groupe de déploiement dans un déploiement individuel.

Vous pouvez également choisir de restaurer un déploiement ayant échoué en redéployant manuellement une révision précédemment déployée.

Dans tous les cas, un déploiement nouveau ou annulé est attribué à son propre identifiant de déploiement. La liste des déploiements que vous pouvez consulter dans la CodeDeploy console indique ceux qui sont le résultat d'un déploiement automatique.

Pour de plus amples informations, veuillez consulter [Redéployez et annulez un déploiement avec CodeDeploy](#).

Déploiements sur une plateforme de calcul Amazon ECS

Cette rubrique fournit des informations sur les composants et le flux de travail des CodeDeploy déploiements qui utilisent la plateforme de calcul Amazon ECS.

Rubriques

- [Avant de commencer un déploiement Amazon ECS](#)
- [Flux de travail de déploiement \(haut niveau\) sur une plateforme de calcul Amazon ECS](#)
- [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#)
- [Téléchargement de la version révisée de votre application](#)
- [Création de vos groupes d'applications et de déploiement](#)
- [Déploiement de la révision de votre application](#)
- [Mettre à jour votre application](#)
- [Déploiements interrompus ou échoués](#)
- [Redéploiements et annulations de déploiements](#)
- [blue/green Déploiements Amazon ECS via AWS CloudFormation](#)

Avant de commencer un déploiement Amazon ECS

Avant de commencer le déploiement d'une application Amazon ECS, vous devez disposer des éléments suivants. Certaines exigences sont spécifiées lorsque vous créez votre groupe de déploiement, tandis que d'autres sont spécifiées dans le AppSpec fichier.

Exigence	Mention
Cluster Amazon ECS	Groupe de déploiement
Amazon ECS service	Groupe de déploiement
Application Load Balancer ou Network Load Balancer	Groupe de déploiement
Écouteur de production	Groupe de déploiement
Écouteur de test (facultatif)	Groupe de déploiement

Exigence	Mention
Deux groupes cibles	Groupe de déploiement
Définition de tâche Amazon ECS	AppSpec fichier
Nom du conteneur	AppSpec fichier
Port du conteneur	AppSpec fichier

Cluster Amazon ECS

Un cluster Amazon ECS est un regroupement logique de tâches ou de services. Vous spécifiez le cluster Amazon ECS qui contient votre service Amazon ECS lorsque vous créez le groupe de déploiement de votre CodeDeploy application. Pour plus d'informations, consultez les [clusters Amazon ECS](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service.

Service Amazon ECS

Un service Amazon ECS gère et exécute des instances spécifiées d'une définition de tâche dans un cluster Amazon ECS. Votre service Amazon ECS doit être activé pour CodeDeploy. Par défaut, un service Amazon ECS est activé pour les déploiements Amazon ECS. Lorsque vous créez votre groupe de déploiement, vous choisissez de déployer un service Amazon ECS qui se trouve dans votre cluster Amazon ECS. Pour plus d'informations, consultez les [services Amazon ECS](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service.

Application Load Balancer ou Network Load Balancer

Vous devez utiliser Elastic Load Balancing avec le service Amazon ECS que vous souhaitez mettre à jour avec un déploiement Amazon ECS. Vous pouvez utiliser un Application Load Balancer ou un Network Load Balancer. Nous recommandons un Application Load Balancer afin que vous puissiez tirer parti de fonctionnalités telles que le mappage dynamique des ports, le routage basé sur le chemin et les règles de priorité. Vous spécifiez l'équilibreur de charge lorsque vous créez le groupe de déploiement de votre CodeDeploy application. Pour plus d'informations, consultez la section [Création Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS d'un équilibreur de charge](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service.

Un ou deux auditeurs

Un écouteur est utilisé par votre équilibreur de charge pour diriger le trafic vers vos groupes cibles. Un écouteur de production est obligatoire. Vous pouvez spécifier un deuxième écouteur de test facultatif qui dirige le trafic vers votre ensemble de tâches de remplacement lors de l'exécution des tests de validation. Vous spécifiez un écouteur ou les deux lors de la création de votre groupe de déploiement. Si vous utilisez la console Amazon ECS pour créer votre service Amazon ECS, vos écouteurs sont créés pour vous. Pour plus d'informations, consultez les sections [Listeners for your application load balancers](#) dans le guide de l'utilisateur d'Elastic Load Balancing et [Creating a service](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service.

Deux groupes cibles Amazon ECS

Un groupe cible est utilisé pour acheminer le trafic vers une cible enregistrée. Un déploiement Amazon ECS nécessite deux groupes cibles : un pour l'ensemble de tâches d'origine de votre application Amazon ECS et un pour son ensemble de tâches de remplacement. Au cours du déploiement, CodeDeploy crée un ensemble de tâches de remplacement et redirige le trafic de l'ensemble de tâches d'origine vers le nouveau. Vous spécifiez les groupes cibles lors de la création du groupe de déploiement de votre application CodeDeploy .

Au cours d'un déploiement, CodeDeploy déterminez quel groupe cible est associé à l'ensemble de tâches de votre service Amazon ECS qui possède le statut PRIMARY (il s'agit de l'ensemble de tâches d'origine) et associe un groupe cible à celui-ci, puis associe l'autre groupe cible à l'ensemble de tâches de remplacement. Si vous effectuez un autre déploiement, le groupe cible associé à l'ensemble de tâches d'origine du déploiement actuel est associé à l'ensemble de tâches de remplacement du déploiement suivant. Pour plus d'informations, consultez la section [Groupes cibles pour les équilibreurs de charge de votre application dans le guide de l'utilisateur d'Elastic Load Balancing](#).

Une définition de tâche Amazon ECS

Une définition de tâche est requise pour exécuter le conteneur Docker qui contient votre application Amazon ECS. Vous spécifiez l'ARN de votre définition de tâche dans le AppSpec fichier de votre CodeDeploy application. Pour plus d'informations, consultez les [définitions des tâches Amazon ECS](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service et [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

Un conteneur pour votre application Amazon ECS

Un conteneur Docker est une unité de logiciel qui regroupe le code et ses dépendances afin que votre application puisse s'exécuter. Un conteneur isole votre application afin qu'elle s'exécute

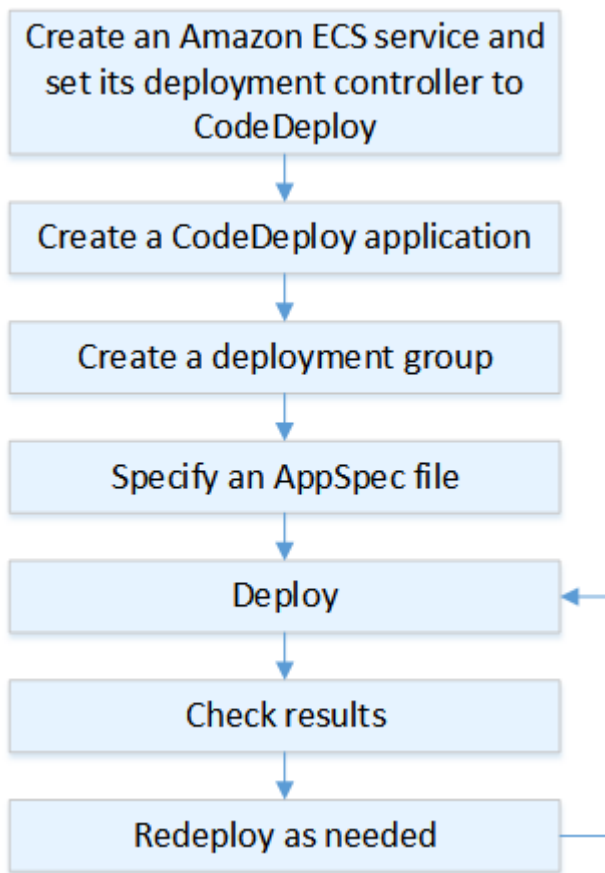
dans différents environnements informatiques. Votre équilibreur de charge dirige le trafic vers un conteneur dans l'ensemble de tâches de votre application Amazon ECS. Vous spécifiez le nom de votre conteneur dans le AppSpec fichier de votre CodeDeploy application. Le conteneur spécifié dans votre AppSpec fichier doit être l'un des conteneurs spécifiés dans votre définition de tâche Amazon ECS. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon Elastic Container Service ?](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service et [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

Un port pour votre ensemble de tâches de remplacement

Lors de votre déploiement Amazon ECS, votre équilibreur de charge dirige le trafic vers ce port sur le conteneur spécifié dans le AppSpec fichier de votre CodeDeploy application. Vous spécifiez le port dans le AppSpec fichier de votre CodeDeploy application. Pour de plus amples informations, veuillez consulter [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

Flux de travail de déploiement (haut niveau) sur une plateforme de calcul Amazon ECS

Le schéma suivant montre les principales étapes du déploiement des services Amazon ECS mis à jour.



Les étapes sont les suivantes :

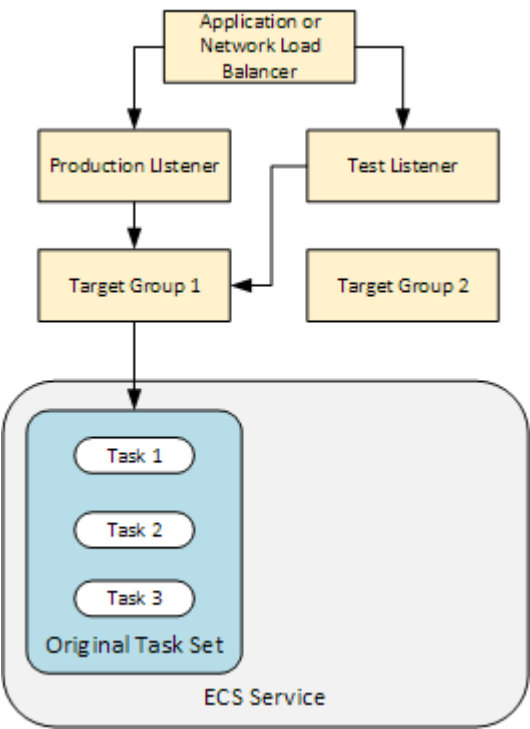
1. Créez une AWS CodeDeploy application en spécifiant un nom qui représente de manière unique ce que vous souhaitez déployer. Pour déployer une application Amazon ECS, dans votre AWS CodeDeploy application, choisissez la plateforme de calcul Amazon ECS. CodeDeploy utilise une application lors d'un déploiement pour référencer les composants de déploiement appropriés, tels que le groupe de déploiement, les groupes cibles, les écouteurs, le comportement de réacheminement du trafic et la révision de l'application. Pour de plus amples informations, veuillez consulter [Créez une application avec CodeDeploy](#).
2. Configurez un groupe de déploiement en spécifiant les éléments suivants :
 - Le nom du groupe de déploiement.
 - Votre cluster Amazon ECS et le nom de votre service. Le contrôleur de déploiement du service Amazon ECS doit être configuré sur CodeDeploy.
 - L'écouteur de production, un écouteur de test facultatif et les groupes cibles utilisés au cours d'un déploiement.

- Les paramètres de déploiement, tels que le moment où le trafic de production doit être redirigé vers le jeu de tâches Amazon ECS de remplacement dans votre service Amazon ECS et le moment où vous devez mettre fin à l'ensemble de tâches Amazon ECS d'origine dans votre service Amazon ECS.
 - Des paramètres facultatifs, tels que les déclencheurs, les alarmes et le comportement de restauration.
3. Spécifiez un fichier de spécification d'application (AppSpec fichier). Vous pouvez le télécharger sur Amazon S3, le saisir dans la console au format YAML ou JSON, ou le spécifier à l'aide du AWS CLI SDK. Le AppSpec fichier spécifie une définition de tâche Amazon ECS pour le déploiement, le nom du conteneur et le mappage des ports utilisés pour acheminer le trafic, ainsi que les fonctions Lambda exécutées après les hooks du cycle de vie du déploiement. Le nom du conteneur doit être un conteneur dans votre définition de tâche Amazon ECS. Pour de plus amples informations, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).
 4. Déployez la révision de votre application. AWS CodeDeploy redirige le trafic de la version d'origine d'un ensemble de tâches de votre service Amazon ECS vers un nouvel ensemble de tâches de remplacement. Les groupes cibles spécifiés dans le groupe de déploiement sont utilisés pour diriger le trafic vers les ensembles de tâches d'origine et de remplacement. Lorsque le déploiement est terminé, l'ensemble de tâches d'origine est arrêté. Vous pouvez spécifier un écouteur de test facultatif pour diriger le trafic de test vers votre version de remplacement avant que le trafic soit réacheminé vers celle-ci. Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).
 5. Vérifiez les résultats du déploiement. Pour de plus amples informations, veuillez consulter [Surveillance des déploiements dans CodeDeploy](#).

Que se passe-t-il lors d'un déploiement d'Amazon ECS

Avant de démarrer un déploiement Amazon ECS avec un écouteur de test, vous devez configurer ses composants. Pour de plus amples informations, veuillez consulter [Avant de commencer un déploiement Amazon ECS](#).

Le schéma suivant montre la relation entre ces composants lorsqu'un déploiement Amazon ECS est prêt à démarrer.



Lorsque le déploiement démarre, les événements de cycle de vie du déploiement s'exécutent les uns après les autres. Certains événements du cycle de vie sont des hooks qui exécutent uniquement les fonctions Lambda spécifiées dans le AppSpec fichier. Dans le tableau suivant, les événements de cycle de vie du déploiement sont répertoriés dans leur ordre d'exécution. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement Amazon ECS](#).

Événement lié au cycle	Action liée aux événements liés au cycle
BeforeInstall (un crochet pour les fonctions Lambda)	Exécutez les fonctions Lambda.
Installation	Configuration de l'ensemble de tâches de remplacement.
AfterInstall (un crochet pour les fonctions Lambda)	Exécutez les fonctions Lambda.
AllowTestTraffic	Acheminement du trafic de l'écouteur de test au groupe cible 2.

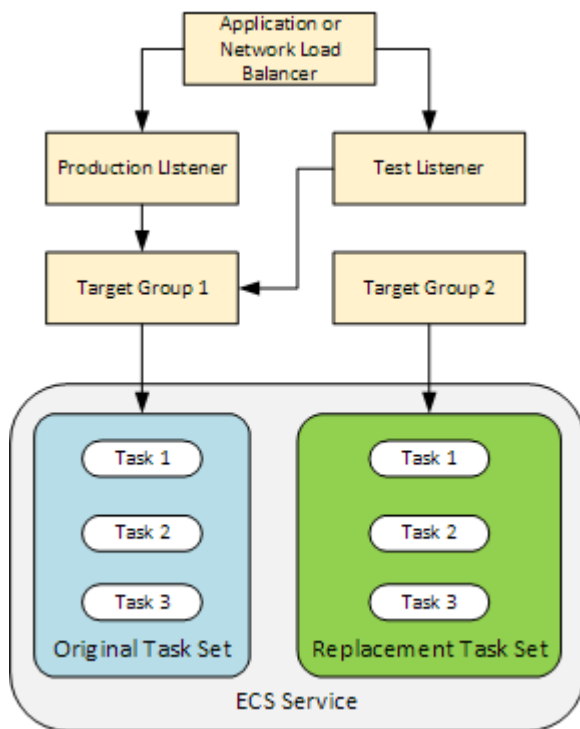
Événement lié au cycle	Action liée aux événements liés au cycle
<code>AfterAllowTestTraffic</code> (un crochet pour les fonctions Lambda)	Exécutez les fonctions Lambda.
<code>BeforeAllowTraffic</code> (un crochet pour les fonctions Lambda)	Exécutez les fonctions Lambda.
<code>AllowTraffic</code>	Acheminement du trafic de l'écouteur de production au groupe cible 2.
<code>AfterAllowTraffic</code>	Exécutez les fonctions Lambda.

 Note

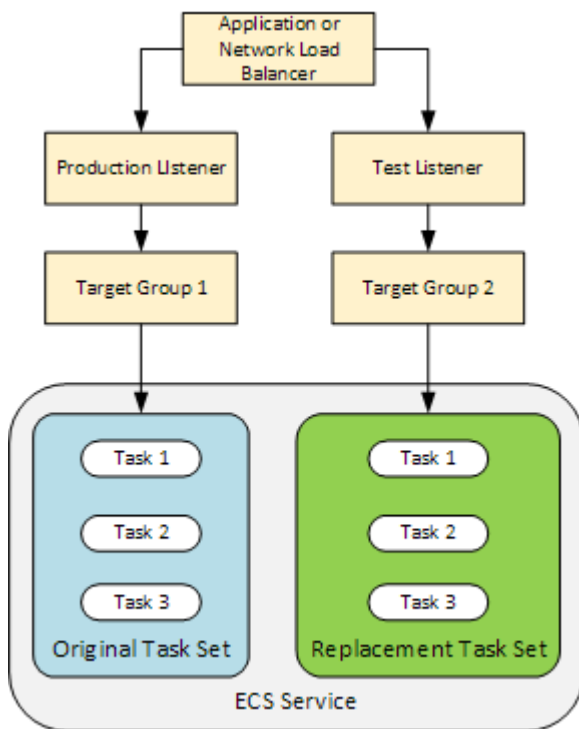
Les fonctions Lambda d'un hook sont facultatives.

1. Exécute toutes les fonctions Lambda spécifiées dans le `BeforeInstall` hook du fichier `AppSpec`
2. Au cours de l'événement de cycle de vie `Install` :
 - a. Un ensemble de tâches de remplacement est créé dans votre service Amazon ECS.
 - b. L'application conteneurisée mise à jour est installée dans l'ensemble de tâches de remplacement.
 - c. Le deuxième groupe cible est associé à l'ensemble de tâches de remplacement.

Ce schéma présente les composants de déploiement avec le nouvel ensemble de tâches de remplacement. L'application conteneurisée est à l'intérieur de cet ensemble de tâches. L'ensemble de tâches se compose de trois tâches. (Une application peut comporter n'importe quel nombre de tâches.) Le deuxième groupe cible est désormais associé à l'ensemble de tâches de remplacement.



3. Exécute toutes les fonctions Lambda spécifiées dans le `AfterInstall` hook du fichier `AppSpec`
4. L'événement `AllowTestTraffic` est appelé. Durant cet événement de cycle de vie, l'écouteur de test achemine le trafic vers l'application conteneurisée mise à jour.



5.

Exécute toutes les fonctions Lambda spécifiées dans le `AfterAllowTestTraffic` hook du fichier. AppSpec Les fonctions Lambda peuvent valider le déploiement à l'aide du trafic de test. Par exemple, une fonction Lambda peut diriger le trafic vers l'écouteur de test et suivre les métriques de l'ensemble de tâches de remplacement. Si des annulations sont configurées, vous pouvez configurer une CloudWatch alarme qui déclenche une annulation lorsque le test de validation de votre fonction Lambda échoue.

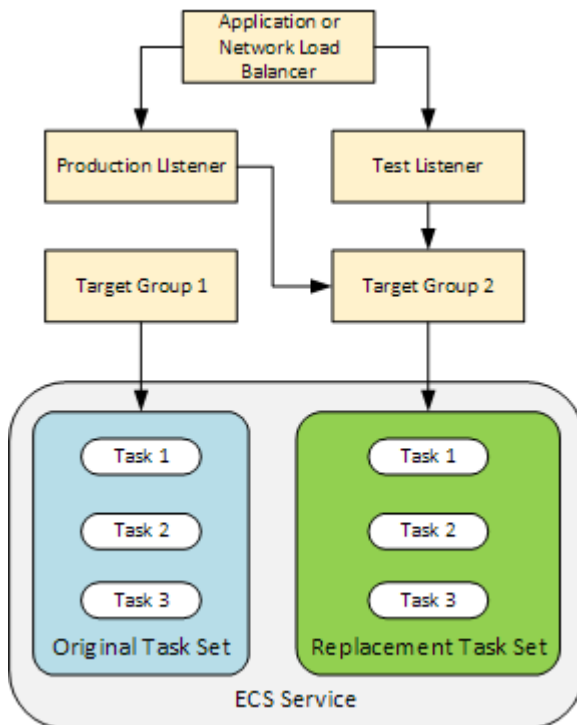
Une fois les tests de validation terminés, l'une des actions suivantes se produit :

- Si la validation échoue et que des restaurations sont configurées, le statut du déploiement est défini sur `Failed` et les composants reviennent à l'état dans lequel ils étaient au démarrage du déploiement.
- Si la validation échoue et qu'aucune restauration n'est configurée, le statut du déploiement est défini sur `Failed` et les composants restent dans leur état actuel.
- Si la validation aboutit, le déploiement se poursuit avec le hook `BeforeAllowTraffic`.

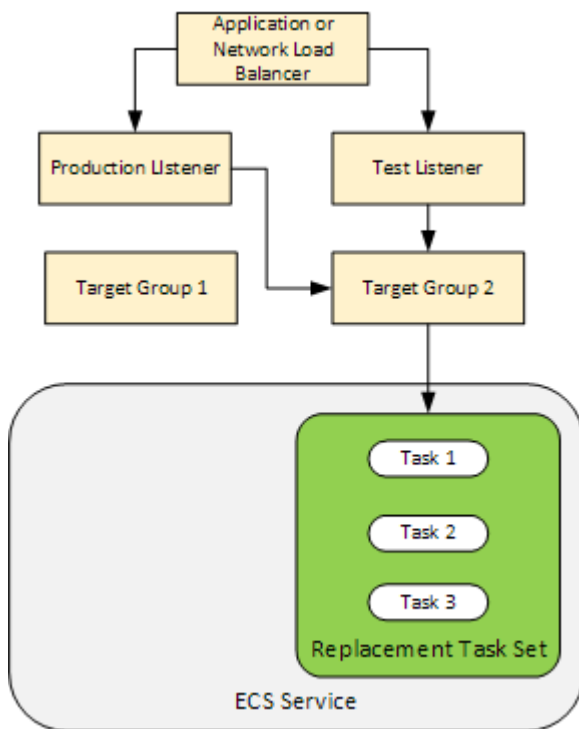
Pour plus d'informations, consultez [Surveillance des déploiements avec des CloudWatch alarmes dans CodeDeploy](#), [Annulations automatiques](#) et [Configuration des options avancées d'un groupe de déploiement](#).

6. Exécute toutes les fonctions Lambda spécifiées dans le `BeforeAllowTraffic` hook du fichier `AppSpec`

7. L'événement `AllowTraffic` est appelé. Le trafic de production est redirigé de l'ensemble de tâches d'origine vers l'ensemble de tâches de remplacement. Sur le schéma suivant, l'ensemble de tâches de remplacement reçoit du trafic de production.



8. Exécute toutes les fonctions Lambda spécifiées dans le `AfterAllowTraffic` hook du fichier `AppSpec`
9. Une fois tous les événements réussis, le statut du déploiement est défini sur `Succeeded` et l'ensemble de tâches d'origine est supprimé.



Téléchargement de la version révisée de votre application

Placez un AppSpec fichier dans Amazon S3 ou saisissez-le directement dans la console ou AWS CLI. Pour de plus amples informations, veuillez consulter [Application Specification Files](#).

Création de vos groupes d'applications et de déploiement

Un groupe de CodeDeploy déploiement sur une plate-forme de calcul Amazon ECS identifie les écouteurs destinés à acheminer le trafic vers votre application Amazon ECS mise à jour et les deux groupes cibles utilisés lors de votre déploiement. Un groupe de déploiement définit également un ensemble d'options de configuration, notamment les alarmes et les configurations de restauration.

Déploiement de la révision de votre application

Vous êtes maintenant prêt à déployer le service Amazon ECS mis à jour spécifié dans votre groupe de déploiement. Vous pouvez utiliser la CodeDeploy console ou la commande [create-deployment](#). Vous pouvez spécifier certains paramètres pour contrôler votre déploiement, y compris la révision et le groupe de déploiement.

Mettre à jour votre application

Vous pouvez mettre à jour votre application, puis utiliser la CodeDeploy console ou appeler la commande [create-deployment](#) pour envoyer une révision.

Déploiements interrompus ou échoués

Vous pouvez utiliser la CodeDeploy console ou la commande [stop-deployment](#) pour arrêter un déploiement. Lorsque vous essayez d'arrêter le déploiement, trois choses peuvent se produire :

- Le déploiement s'arrête et l'opération renvoie un statut de réussite. Dans ce cas, aucun événement de cycle de vie de déploiement supplémentaire ne se déroule sur le groupe de déploiement pour le déploiement arrêté.
- Le déploiement ne s'arrête pas immédiatement et l'opération renvoie un statut d'attente. Dans ce cas, certains événements de cycle de vie de déploiement peuvent encore se dérouler sur le groupe de déploiement. Une fois l'opération en attente terminée, les appels suivants visant à arrêter le déploiement renvoient un statut de réussite.
- Le déploiement ne peut pas s'arrêter et l'opération renvoie une erreur. Pour plus d'informations, consultez les sections [Informations sur les erreurs](#) et [Erreurs courantes](#) dans le Guide de référence de l' AWS CodeDeploy API.

Redéploiements et annulations de déploiements

CodeDeploy implémente les annulations en redirigeant le trafic de l'ensemble de tâches de remplacement vers l'ensemble de tâches d'origine.

Vous pouvez configurer un groupe de déploiement pour restaurer automatiquement les déploiements lorsque certaines conditions sont remplies, y compris lorsqu'un déploiement échoue, ou si un seuil de surveillance d'alarme est atteint. Vous pouvez également remplacer les paramètres de restauration spécifiés pour un groupe de déploiement dans un déploiement individuel.

Vous pouvez également choisir de restaurer un déploiement ayant échoué en redéployant manuellement une révision précédemment déployée.

Dans tous les cas, un déploiement nouveau ou annulé est attribué à son propre identifiant de déploiement. La console CodeDeploy affiche une liste des déploiements qui résultent d'un déploiement automatique.

Si vous effectuez un autre déploiement, le groupe cible associé à l'ensemble de tâches d'origine du déploiement actuel est associé à l'ensemble de tâches de remplacement du déploiement.

Pour de plus amples informations, veuillez consulter [Redéployez et annulez un déploiement avec CodeDeploy](#).

blue/green Déploiements Amazon ECS via AWS CloudFormation

Vous pouvez l'utiliser AWS CloudFormation pour gérer les blue/green déploiements Amazon ECS via CodeDeploy. Pour de plus amples informations, veuillez consulter [Créez un blue/green déploiement Amazon ECS via CloudFormation](#).

Note

La gestion des blue/green déploiements Amazon ECS avec n' CloudFormation est pas disponible dans la région Asie-Pacifique (Osaka).

Déploiements sur une plateforme informatique EC2/sur site

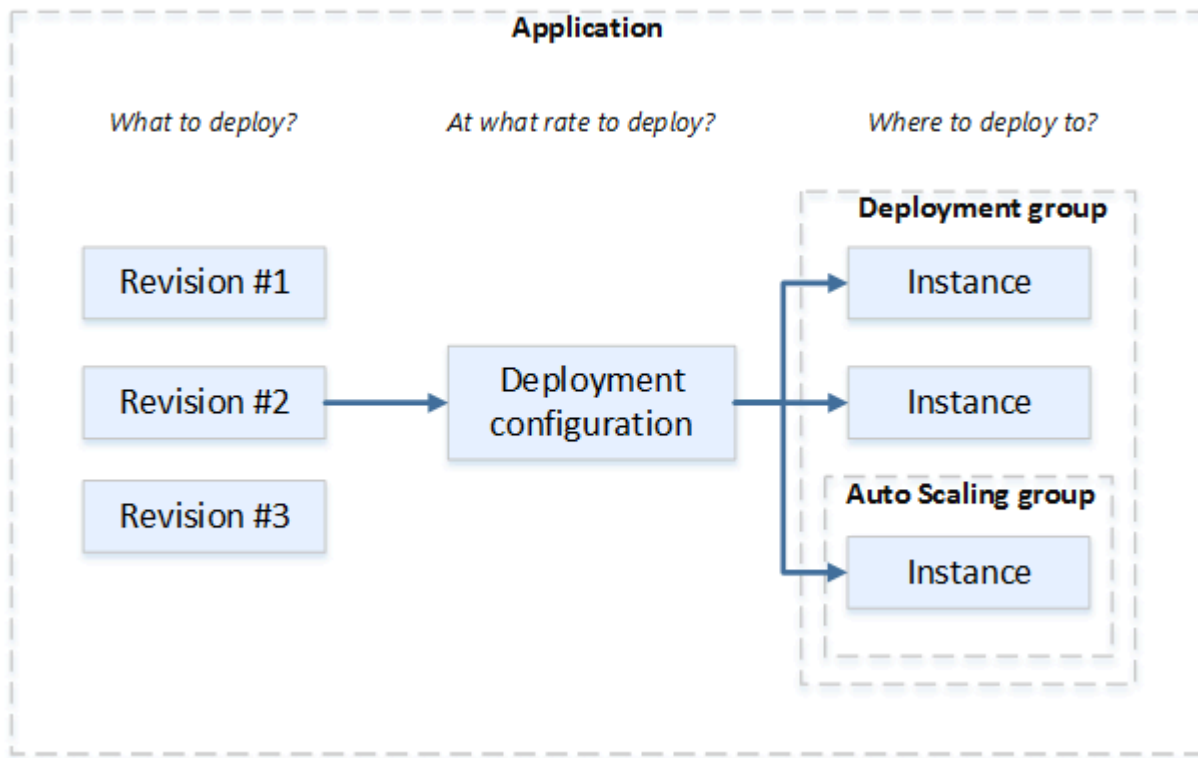
Cette rubrique fournit des informations sur les composants et le flux de travail des CodeDeploy déploiements qui utilisent la plate-forme de calcul EC2/on-premises. Pour plus d'informations sur blue/green les déploiements, consultez [Vue d'ensemble d'un blue/green déploiement](#).

Rubriques

- [Composants de déploiement sur une plate-forme informatique EC2/sur site](#)
- [Flux de travail de déploiement sur une plate-forme informatique EC2/sur site](#)
- [Configuration des instances](#)
- [Téléchargement de la version révisée de votre application](#)
- [Création de vos groupes d'applications et de déploiement](#)
- [Déploiement de la révision de votre application](#)
- [Mettre à jour votre application](#)
- [Déploiements interrompus ou échoués](#)
- [Redéploiements et annulations de déploiements](#)

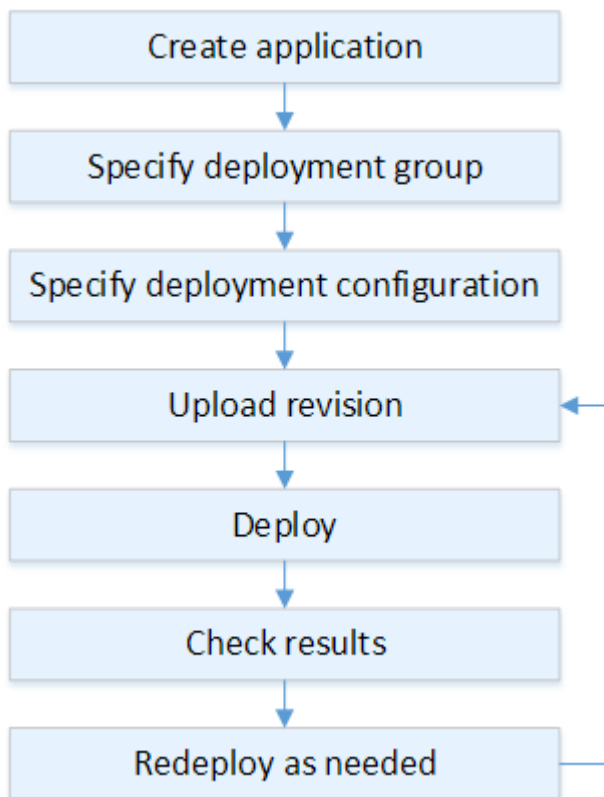
Composants de déploiement sur une plate-forme informatique EC2/sur site

Le schéma suivant montre les composants d'un CodeDeploy déploiement sur une plate-forme informatique EC2/sur site.



Flux de travail de déploiement sur une plate-forme informatique EC2/sur site

Le schéma ci-dessous illustre les principales étapes du déploiement de révisions d'application :



Les étapes sont les suivantes :

1. Créez une application et donnez-lui un nom qui identifie de manière unique les révisions de l'application que vous souhaitez déployer et la plate-forme de calcul de votre application. CodeDeploy utilise ce nom lors d'un déploiement pour s'assurer qu'il fait référence aux composants de déploiement appropriés, tels que le groupe de déploiement, la configuration du déploiement et la révision de l'application. Pour de plus amples informations, veuillez consulter [Créez une application avec CodeDeploy](#).
2. Configurez un groupe de déploiement en spécifiant un type de déploiement et les instances sur lesquelles vous souhaitez déployer vos révisions d'application. Un déploiement sur place met à jour les instances avec la révision d'application la plus récente. Un blue/green déploiement enregistre un ensemble d'instances de remplacement pour le groupe de déploiement auprès d'un équilibreur de charge et annule l'enregistrement des instances d'origine.

Vous pouvez spécifier les balises appliquées aux instances, les noms des groupes Amazon EC2 Auto Scaling, ou les deux.

Si vous spécifiez un groupe de balises dans un groupe de déploiement, il est CodeDeploy déployé sur les instances auxquelles au moins une des balises spécifiées est appliquée. Si vous spécifiez

au moins deux groupes de balises, le CodeDeploy déploiement s'effectue uniquement sur les instances qui répondent aux critères de chacun des groupes de balises. Pour de plus amples informations, veuillez consulter [Tagging Instances for Deployments](#).

Dans tous les cas, les instances doivent être configurées pour être utilisées dans un déploiement (c'est-à-dire qu'elles doivent être étiquetées ou appartenir à un groupe Amazon EC2 Auto Scaling) et que CodeDeploy l'agent doit être installé et exécuté.

Nous vous fournissons un CloudFormation modèle que vous pouvez utiliser pour configurer rapidement une instance Amazon EC2 basée sur Amazon Linux ou Windows Server. Nous vous fournissons également l' CodeDeploy agent autonome afin que vous puissiez l'installer sur des instances Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) ou Windows Server. Pour de plus amples informations, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#).

Vous pouvez également spécifier les options suivantes :

- Notifications Amazon SNS. Créez des déclencheurs qui envoient des notifications aux abonnés d'une rubrique Amazon SNS lorsque des événements spécifiques, tels que des événements de réussite ou d'échec, se produisent dans les déploiements et les instances. Pour de plus amples informations, veuillez consulter [Monitoring Deployments with Amazon SNS Event Notifications](#).
 - Gestion de déploiement basée sur des alarmes. Mettez en œuvre la surveillance des CloudWatch alarmes Amazon pour arrêter les déploiements lorsque vos indicateurs dépassent ou tombent en CloudWatch dessous des seuils définis.
 - Restaurations de déploiement automatiques. Configurez un déploiement pour que la dernière révision correcte connue soit restaurée automatiquement lorsqu'un déploiement échoue ou si un seuil d'alarme est atteint.
3. Spécifiez une configuration de déploiement pour indiquer le nombre d'instances sur lesquelles déployer simultanément vos révisions d'application et en décrivant les conditions de réussite et d'échec du déploiement. Pour de plus amples informations, veuillez consulter [View Deployment Configuration Details](#).
 4. Téléchargez une révision de l'application sur Amazon S3 ou GitHub. Outre les fichiers que vous souhaitez déployer et les scripts que vous souhaitez exécuter pendant le déploiement, vous devez inclure un fichier de spécification de l'application (AppSpec fichier). Ce fichier contient les instructions de déploiement, indiquant par exemple où copier les fichiers sur chaque instance et à quel moment exécuter les scripts de déploiement. Pour de plus amples informations, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

5. Déployez votre révision d'application dans le groupe de déploiement. L' CodeDeploy agent de chaque instance du groupe de déploiement copie la révision de votre application depuis Amazon S3 ou GitHub vers l'instance. L' CodeDeploy agent déregroupe ensuite la révision et, à l'aide du AppSpec fichier, copie les fichiers dans les emplacements spécifiés et exécute tous les scripts de déploiement. Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).
6. Vérifiez les résultats du déploiement. Pour de plus amples informations, veuillez consulter [Surveillance des déploiements dans CodeDeploy](#).
7. Redéployez une révision. Vous pouvez effectuer cette opération si vous avez besoin de corriger un bogue dans le contenu source ou d'exécuter les scripts de déploiement dans un ordre différent, ou encore de réagir à un échec de déploiement. Pour ce faire, regroupez votre contenu source révisé, tous les scripts de déploiement et le AppSpec fichier dans une nouvelle révision, puis chargez la révision dans le compartiment ou le GitHub référentiel Amazon S3. Vous exécutez alors un nouveau déploiement dans le même groupe de déploiement avec la nouvelle révision. Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).

Configuration des instances

Vous devez configurer les instances avant de déployer des révisions d'application pour la première fois. Si une révision d'application a besoin de trois serveurs de production et de deux serveurs de sauvegarde, vous devez lancer ou utiliser cinq instances.

Pour mettre en service manuellement les instances :

1. Installez l' CodeDeploy agent sur les instances. L' CodeDeploy agent peut être installé sur des instances Amazon Linux, Ubuntu Server, RHEL et Windows Server.
2. Activez le balisage si vous utilisez des balises pour identifier les instances d'un groupe de déploiement. CodeDeploy s'appuie sur des balises pour identifier et regrouper les instances dans des groupes CodeDeploy de déploiement. Les didacticiels de mise en route utilisaient les deux, mais vous pouvez utiliser simplement une clé ou une valeur pour définir une balise pour un groupe de déploiement.
3. Lancez des instances Amazon EC2 associées à un profil d'instance IAM. Le profil d'instance IAM doit être attaché à une instance Amazon EC2 lors de son lancement pour que CodeDeploy l'agent vérifie l'identité de l'instance.
4. Créez un rôle de service. Fournissez un accès au service CodeDeploy afin d'étendre le nombre de balises de votre AWS compte.

Pour un déploiement initial, le CloudFormation modèle fait tout cela pour vous. Il crée et configure de nouvelles instances Amazon EC2 uniques basées sur Amazon Linux ou Windows Server avec CodeDeploy l'agent déjà installé. Pour de plus amples informations, veuillez consulter [Utilisation d'instances pour CodeDeploy](#).

 Note

Pour un blue/green déploiement, vous pouvez choisir entre utiliser les instances dont vous disposez déjà pour l'environnement de remplacement ou laisser de CodeDeploy nouvelles instances vous être fournies dans le cadre du processus de déploiement.

Téléchargement de la version révisée de votre application

Placez un AppSpec fichier sous le dossier racine dans la structure de dossiers de contenu source de votre application. Pour de plus amples informations, veuillez consulter [Application Specification Files](#).

Groupez la structure de dossiers de contenu source de l'application dans un format de fichier d'archive tel que zip, tar ou tar compressé. Téléchargez le fichier d'archive (la révision) dans un compartiment ou un GitHub référentiel Amazon S3.

 Note

Les formats de fichier d'archive tar et tar compressé (.tar et .tar.gz) ne sont pas pris en charge pour les instances Windows Server.

Création de vos groupes d'applications et de déploiement

Un groupe de CodeDeploy déploiement identifie un ensemble d'instances en fonction de leurs balises, des noms de groupes Amazon EC2 Auto Scaling, ou des deux. Plusieurs révisions d'application peuvent être déployées sur la même instance. Une révision d'application peut être déployée sur plusieurs instances.

Par exemple, vous pouvez ajouter une balise « Prod » aux trois serveurs de production et une balise « Sauvegarde » aux deux serveurs de sauvegarde. Ces deux balises peuvent être utilisées pour créer deux groupes de déploiement différents dans l' CodeDeploy application, ce qui vous permet de choisir quel ensemble de serveurs (ou les deux) doit participer à un déploiement.

Vous pouvez utiliser plusieurs groupes de balises dans un groupe de déploiement pour restreindre les déploiements à un plus petit ensemble d'instances. Pour plus d'informations, consultez [Tagging Instances for Deployments](#).

Déploiement de la révision de votre application

Vous êtes maintenant prêt à déployer la révision de votre application depuis Amazon S3 ou GitHub vers le groupe de déploiement. Vous pouvez utiliser la CodeDeploy console ou la commande [create-deployment](#). Vous pouvez spécifier certains paramètres pour contrôler votre déploiement, y compris la révision, le groupe de déploiement et la configuration de déploiement.

Mettre à jour votre application

Vous pouvez mettre à jour votre application, puis utiliser la CodeDeploy console ou appeler la commande [create-deployment](#) pour envoyer une révision.

Déploiements interrompus ou échoués

Vous pouvez utiliser la CodeDeploy console ou la commande [stop-deployment](#) pour arrêter un déploiement. Lorsque vous essayez d'arrêter le déploiement, trois choses peuvent se produire :

- Le déploiement s'arrête et l'opération renvoie un statut de réussite. Dans ce cas, aucun événement de cycle de vie de déploiement supplémentaire ne se déroule sur le groupe de déploiement pour le déploiement arrêté. Certains fichiers peuvent déjà avoir été copiés sur une ou plusieurs des instances figurant dans le groupe de déploiement, et certains scripts peuvent déjà y avoir été exécutés.
- Le déploiement ne s'arrête pas immédiatement et l'opération renvoie un statut d'attente. Dans ce cas, certains événements de cycle de vie de déploiement peuvent encore se dérouler sur le groupe de déploiement. Certains fichiers peuvent déjà avoir été copiés sur une ou plusieurs des instances figurant dans le groupe de déploiement, et certains scripts peuvent déjà y avoir été exécutés. Une fois l'opération en attente terminée, les appels suivants visant à arrêter le déploiement renvoient un statut de réussite.
- Le déploiement ne peut pas s'arrêter et l'opération renvoie une erreur. Pour plus d'informations, consultez [ErrorInformation](#) la section [Erreurs courantes](#) dans la référence AWS CodeDeploy d'API.

Comme les déploiements arrêtés, les déploiements ayant échoué peuvent se traduire par des événements de cycle de vie de déploiement ayant déjà été exécutés sur une ou plusieurs des instances figurant dans le groupe de déploiement. Pour savoir pourquoi un déploiement a échoué,

vous pouvez utiliser la CodeDeploy console, appeler la [get-deployment-instance](#) commande ou analyser les données du fichier journal de l'échec du déploiement. Pour plus d'informations, consultez [Révision de l'application et nettoyage des fichiers journaux](#) et [Afficher les données du journal pour les déploiements CodeDeploy EC2/sur site](#).

Redéploiements et annulations de déploiements

CodeDeploy implémente les annulations en redéployant, en tant que nouveau déploiement, une révision précédemment déployée.

Vous pouvez configurer un groupe de déploiement pour restaurer automatiquement les déploiements lorsque certaines conditions sont remplies, y compris lorsqu'un déploiement échoue, ou si un seuil de surveillance d'alarme est atteint. Vous pouvez également remplacer les paramètres de restauration spécifiés pour un groupe de déploiement dans un déploiement individuel.

Vous pouvez également choisir de restaurer un déploiement ayant échoué en redéployant manuellement une révision précédemment déployée.

Dans tous les cas, un déploiement nouveau ou annulé est attribué à son propre identifiant de déploiement. La liste des déploiements que vous pouvez consulter dans la CodeDeploy console indique ceux qui sont le résultat d'un déploiement automatique.

Pour de plus amples informations, veuillez consulter [Redéployez et annulez un déploiement avec CodeDeploy](#).

CodeDeploy fichiers de spécification de l'application (AppSpec)

Un fichier de spécification d'application (AppSpec fichier), unique à CodeDeploy, est un fichier au format [YAML](#) ou au format [JSON](#). Le AppSpec fichier est utilisé pour gérer chaque déploiement sous la forme d'une série de hooks d'événements du cycle de vie, définis dans le fichier.

Pour plus d'informations sur la création d'un AppSpec fichier correctement formé, consultez [CodeDeploy AppSpec référence de fichier](#).

Rubriques

- [AppSpec fichiers sur une plateforme de calcul Amazon ECS](#)
- [AppSpec fichiers sur une plate-forme AWS Lambda informatique](#)
- [AppSpec fichiers sur une plate-forme informatique EC2/sur site](#)

- [Comment l' CodeDeploy agent utilise le AppSpec fichier](#)

AppSpec fichiers sur une plateforme de calcul Amazon ECS

Si votre application utilise la plate-forme de calcul Amazon ECS, le AppSpec fichier peut être formaté en YAML ou en JSON. Il peut également être directement entré dans un éditeur via la console. Le AppSpec fichier est utilisé pour spécifier :

- Le nom du service Amazon ECS ainsi que le nom du conteneur et le port utilisés pour diriger le trafic vers le nouvel ensemble de tâches.
- Les fonctions à utiliser comme tests de validation.

Vous pouvez exécuter des fonctions Lambda de validation après des événements du cycle de vie du déploiement. Pour plus d'informations, consultez [AppSpec section « hooks » pour un déploiement Amazon ECS](#), [AppSpec structure de fichiers pour les déploiements Amazon ECS](#) et [AppSpec Exemple de fichier pour un déploiement Amazon ECS](#).

AppSpec fichiers sur une plate-forme AWS Lambda informatique

Si votre application utilise la plate-forme de calcul AWS Lambda, le AppSpec fichier peut être formaté en YAML ou en JSON. Il peut également être directement entré dans un éditeur via la console. Le AppSpec fichier est utilisé pour spécifier :

- Version de la AWS Lambda fonction à déployer.
- Les fonctions à utiliser comme tests de validation.

Vous pouvez exécuter des fonctions Lambda de validation après des événements du cycle de vie du déploiement. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement AWS Lambda](#).

AppSpec fichiers sur une plate-forme informatique EC2/sur site

Si votre application utilise la plate-forme de calcul EC2/on-premises, le AppSpec fichier est toujours au format YAML. Le AppSpec fichier est utilisé pour :

- mapper les fichiers sources figurant dans votre révision d'application vers leurs destinations sur l'instance ;

- spécifier des autorisations personnalisées pour les fichiers déployés ;
- spécifier les scripts à exécuter sur chaque instance à diverses étapes du processus de déploiement.

Vous pouvez exécuter des scripts sur une instance après de nombreux événements du cycle de vie de déploiement individuels. CodeDeploy exécute uniquement les scripts spécifiés dans le fichier, mais ces scripts peuvent appeler d'autres scripts sur l'instance. Vous pouvez exécuter tout type de script tant qu'il est pris en charge par le système d'exploitation en cours d'exécution sur les instances. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement EC2/sur site](#).

Comment l' CodeDeploy agent utilise le AppSpec fichier

Pendant le déploiement, l' CodeDeploy agent recherche le nom de l'événement en cours dans la section hooks du AppSpec fichier. Si l'événement est introuvable, l'agent CodeDeploy passe à l'étape suivante. Si l'événement est détecté, l' CodeDeploy agent récupère la liste des scripts à exécuter. Les scripts sont exécutés de manière séquentielle, dans l'ordre dans lequel ils apparaissent dans le fichier. L'état de chaque script est enregistré dans le fichier journal de l' CodeDeploy agent sur l'instance.

Si un script s'exécute avec succès, il retourne un code de sortie de 0 (zéro).

Note

L' CodeDeploy agent n'est pas utilisé dans un AWS déploiement Lambda ou Amazon ECS.

Lors de l'événement d'installation, l' CodeDeploy agent utilise les mappages définis dans la section des fichiers du AppSpec fichier pour déterminer les dossiers ou les fichiers à copier de la révision vers l'instance.

Si l' CodeDeploy agent installé sur le système d'exploitation ne correspond pas à ce qui est indiqué dans le AppSpec fichier, le déploiement échoue.

Pour plus d'informations sur les fichiers journaux de l' CodeDeploy agent, consultez [Travailler avec l' CodeDeploy agent](#).

Commencer avec CodeDeploy

Rubriques

- [Étape 1 : Configuration](#)
- [Étape 2 : créer un rôle de service pour CodeDeploy](#)
- [Étape 3 : Limiter les autorisations de l' CodeDeploy utilisateur](#)
- [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#)

Étape 1 : Configuration

Avant de l'utiliser AWS CodeDeploy pour la première fois, vous devez suivre les étapes de configuration. Les étapes consistent à créer un AWS compte (si vous n'en avez pas déjà un) et un utilisateur administratif doté d'un accès programmatique.

Dans ce guide, l'utilisateur administratif est appelé utilisateur CodeDeploy administratif.

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'<https://portal.aws.amazon.com/billing/inscription>.
2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les [tâches nécessitant un accès utilisateur racine](#).

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à <https://aws.amazon.com/> et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisez racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

Pour obtenir des instructions, consultez la section [Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root \(console\)](#) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir [Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center](#) dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section [Connexion au portail AWS d'accès](#) dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

Pour obtenir des instructions, consultez [Création d'un ensemble d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

Pour obtenir des instructions, consultez [Ajout de groupes](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

Vous avez maintenant créé et connecté en tant qu'utilisateur CodeDeploy administratif.

Octroi d'un accès par programmation

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
IAM	(Recommandé) Utilisez les informations d'identification de la console comme informations d'identification temporaires pour signer les demandes programmatiques adressées	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Connexion pour le développement AWS local

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
	au AWS CLI AWS SDKs, ou AWS APIs.	dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, voir Connexion pour le développement AWS local dans le guide de référence AWS SDKs and Tools.
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Configuration du AWS CLI à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, outils, et AWS APIs, voir Authentification IAM Identity Center dans le guide de référence AWS SDKs et Tools.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'AWS Command Line Interface utilisateur. • Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations d'identification à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Important

Nous vous recommandons vivement de configurer l'utilisateur CodeDeploy administratif en tant qu'identité du personnel (un utilisateur géré dans IAM Identity Center) avec le. AWS CLI La plupart des procédures décrites dans ce guide supposent que vous utilisez le AWS CLI pour effectuer des configurations.

 Important

Si vous configurez le AWS CLI, vous serez peut-être invité à spécifier une AWS région. Choisissez l'une des régions prises en charge répertoriées dans [Région et les points de terminaison](#) dans le Références générales AWS.

Étape 2 : créer un rôle de service pour CodeDeploy

Dans AWS, les rôles de service sont utilisés pour accorder des autorisations à un AWS service afin qu'il puisse accéder aux AWS ressources. Les politiques que vous attachez au rôle de service déterminent les ressources auxquelles le service peut accéder et ce qu'il peut faire avec ces ressources.

Le rôle de service pour lequel vous créez CodeDeploy doit disposer des autorisations requises pour votre plate-forme de calcul. Si vous déployez sur plusieurs plateformes informatiques, créez un rôle de service pour chacune d'entre elles. Pour ajouter des autorisations, joignez une ou plusieurs des politiques AWS fournies suivantes :

Pour les déploiements EC2/sur site, joignez la politique. **AWSCodeDeployRole** Elle fournit les autorisations requises pour que votre rôle de service puisse :

- Lisez les balises de vos instances ou identifiez vos instances Amazon EC2 par les noms de groupes Amazon EC2 Auto Scaling.
- Lisez, créez, mettez à jour et supprimez les groupes Amazon EC2 Auto Scaling, les hooks de cycle de vie et les politiques de dimensionnement.
- Publiez des informations dans les rubriques Amazon SNS.
- Récupérez des informations sur les CloudWatch alarmes.
- Lisez et mettez à jour Elastic Load Balancing.

 Note

Si vous créez votre groupe Auto Scaling avec un modèle de lancement, vous devez ajouter les autorisations suivantes :

- `ec2:RunInstances`
- `ec2:CreateTags`

- `iam:PassRole`

Pour plus d'informations [Étape 2 : créer un rôle de service](#), consultez les sections [Création d'un modèle de lancement pour un groupe Auto Scaling](#) et [prise en charge des modèles de lancement](#) dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Pour les déploiements Amazon ECS, si vous souhaitez un accès complet aux services de support, joignez la **AWSCodeDeployRoleForECS** politique. Elle fournit les autorisations requises pour que votre rôle de service puisse :

- Lisez, mettez à jour et supprimez des ensembles de tâches Amazon ECS.
- Mettez à jour les groupes cibles, les auditeurs et les règles d'Elastic Load Balancing.
- Invoquez AWS Lambda des fonctions.
- Accédez aux fichiers de révision dans les compartiments Amazon S3.
- Récupérez des informations sur les CloudWatch alarmes.
- Publiez des informations dans les rubriques Amazon SNS.

Pour les déploiements Amazon ECS, si vous souhaitez un accès limité aux services de support, joignez la **AWSCodeDeployRoleForECSLimited** politique. Elle fournit les autorisations requises pour que votre rôle de service puisse :

- Lisez, mettez à jour et supprimez des ensembles de tâches Amazon ECS.
- Récupérez des informations sur les CloudWatch alarmes.
- Publiez des informations dans les rubriques Amazon SNS.

Pour les déploiements AWS Lambda, si vous souhaitez autoriser la publication sur Amazon SNS, joignez la politique. **AWSCodeDeployRoleForLambda** Elle fournit les autorisations requises pour que votre rôle de service puisse :

- Lisez, mettez à jour et invoquez AWS Lambda des fonctions et des alias.
- Accédez aux fichiers de révision dans les compartiments Amazon S3.
- Récupérez des informations sur les CloudWatch alarmes.
- Publiez des informations dans les rubriques Amazon SNS.

Pour les déploiements AWS Lambda, si vous souhaitez limiter l'accès à Amazon SNS, joignez la politique. **AWSCodeDeployRoleForLambdaLimited** Elle fournit les autorisations requises pour que votre rôle de service puisse :

- Lisez, mettez à jour et invoquez AWS Lambda des fonctions et des alias.
- Accédez aux fichiers de révision dans les compartiments Amazon S3.
- Récupérez des informations sur les CloudWatch alarmes.

Dans le cadre de la configuration du rôle de service, vous mettez à jour également sa relation d'approbation pour spécifier les points de terminaison auxquels vous souhaitez lui accorder l'accès.

Vous pouvez créer un rôle de service avec la console IAM AWS CLI, le ou l' API IAM.

Rubriques

- [Création d'un rôle de service \(console\)](#)
- [Création d'un rôle de service \(CLI\)](#)
- [Obtenir l'ARN du rôle de service \(console\)](#)
- [Obtenez l'ARN du rôle de service \(CLI\)](#)

Création d'un rôle de service (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
3. Choisissez le AWS service, puis sous Cas d'utilisation, dans la liste déroulante, sélectionnez CodeDeploy.
4. Choisissez votre cas d'utilisation :
 - Pour les déploiements EC2/sur site, choisissez. CodeDeploy
 - Pour les déploiements AWS Lambda, choisissez Lambda. CodeDeploy
 - Pour les déploiements Amazon ECS, choisissez CodeDeploy - ECS.
5. Choisissez Suivant.
6. Sur la page Ajouter des autorisations, la politique d'autorisation adaptée au cas d'utilisation est affichée. Choisissez Suivant.

7. Sur la page Nom, révision et création, dans Nom du rôle, entrez le nom du rôle de service (par exemple, **CodeDeployServiceRole**), puis choisissez Créer un rôle.

Vous pouvez également saisir une description de ce rôle de service dans Description du rôle.

8. Si vous souhaitez que ce rôle de service ait l'autorisation d'accéder à tous les points de terminaison actuellement pris en charge, vous avez terminé cette procédure.

Pour empêcher ce rôle de service d'accéder à certains points de terminaison, poursuivez les étapes restantes de cette procédure.

9. Dans la liste des rôles, recherchez et choisissez le rôle que vous venez de créer (CodeDeployServiceRole).
10. Choisissez l'onglet Relations de confiance.
11. Choisissez Modifier la politique d'approbation.

Vous devriez voir la stratégie suivante, qui fournit au rôle de service l'autorisation d'accéder à tous les points de terminaison pris en charge :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pour accorder au rôle de service l'accès à certains points de terminaison pris en charge uniquement, remplacez le contenu de la zone de texte de la politique de confiance par la politique suivante. Supprimez les lignes correspondant aux points de terminaison auxquels vous souhaitez empêcher l'accès, puis choisissez Mettre à jour la politique.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": [

        "codedeploy.us-east-1.amazonaws.com",
        "codedeploy.us-east-2.amazonaws.com",
        "codedeploy.us-west-1.amazonaws.com",
        "codedeploy.us-west-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.ap-east-1.amazonaws.com",
        "codedeploy.ap-northeast-1.amazonaws.com",
        "codedeploy.ap-northeast-2.amazonaws.com",
        "codedeploy.ap-northeast-3.amazonaws.com",
        "codedeploy.ap-southeast-1.amazonaws.com",
        "codedeploy.ap-southeast-2.amazonaws.com",
        "codedeploy.ap-southeast-3.amazonaws.com",
        "codedeploy.ap-southeast-4.amazonaws.com",
        "codedeploy.ap-south-1.amazonaws.com",
        "codedeploy.ap-south-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.eu-west-1.amazonaws.com",
        "codedeploy.eu-west-2.amazonaws.com",
        "codedeploy.eu-west-3.amazonaws.com",
        "codedeploy.eu-central-1.amazonaws.com",
        "codedeploy.eu-central-2.amazonaws.com",
        "codedeploy.eu-north-1.amazonaws.com",
        "codedeploy.eu-south-1.amazonaws.com",
        "codedeploy.eu-south-2.amazonaws.com",
        "codedeploy.il-central-1.amazonaws.com",
        "codedeploy.me-central-1.amazonaws.com",
        "codedeploy.me-south-1.amazonaws.com",
        "codedeploy.sa-east-1.amazonaws.com"

      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

Pour plus d'informations sur la création de rôles de service, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le Guide de l'utilisateur IAM.

Création d'un rôle de service (CLI)

1. Sur votre machine de développement, créez un fichier texte nommé, par exemple, `CodeDeployDemo-Trust.json`. Ce fichier est utilisé pour permettre à CodeDeploy de travailler en votre nom.

Effectuez l'une des actions suivantes :

- Pour accorder l'accès à toutes les AWS régions prises en charge, enregistrez le contenu suivant dans le fichier :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Pour accorder l'accès à certaines régions prises en charge seulement, tapez le contenu suivant dans le fichier et supprimez les lignes pour les régions auxquelles vous souhaitez exclure l'accès :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```

        "codedeploy.us-east-1.amazonaws.com",
        "codedeploy.us-east-2.amazonaws.com",
        "codedeploy.us-west-1.amazonaws.com",
        "codedeploy.us-west-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.ap-east-1.amazonaws.com",
        "codedeploy.ap-northeast-1.amazonaws.com",
        "codedeploy.ap-northeast-2.amazonaws.com",
        "codedeploy.ap-northeast-3.amazonaws.com",
        "codedeploy.ap-southeast-1.amazonaws.com",
        "codedeploy.ap-southeast-2.amazonaws.com",
        "codedeploy.ap-southeast-3.amazonaws.com",
        "codedeploy.ap-southeast-4.amazonaws.com",
        "codedeploy.ap-south-1.amazonaws.com",
        "codedeploy.ap-south-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.eu-west-1.amazonaws.com",
        "codedeploy.eu-west-2.amazonaws.com",
        "codedeploy.eu-west-3.amazonaws.com",
        "codedeploy.eu-central-1.amazonaws.com",
        "codedeploy.eu-central-2.amazonaws.com",
        "codedeploy.eu-north-1.amazonaws.com",
        "codedeploy.eu-south-1.amazonaws.com",
        "codedeploy.eu-south-2.amazonaws.com",
        "codedeploy.il-central-1.amazonaws.com",
        "codedeploy.me-central-1.amazonaws.com",
        "codedeploy.me-south-1.amazonaws.com",
        "codedeploy.sa-east-1.amazonaws.com"
    ],
    },
    "Action": "sts:AssumeRole"
}
]
}

```

Note

N'utilisez pas de virgule après le dernier point de terminaison de la liste.

2. Dans le même répertoire, appelez la commande `create-role` pour créer un rôle de service nommé **CodeDeployServiceRole** sur la base des informations figurant dans le fichier texte que vous venez de créer :

```
aws iam create-role --role-name CodeDeployServiceRole --assume-role-policy-document
file://CodeDeployDemo-Trust.json
```

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

Dans la sortie de la commande, notez la valeur de l'entrée `Arn` sous l'objet `Role`. Vous en aurez besoin plus tard pour créer des groupes de déploiement. Si vous oubliez cette valeur, suivez les instructions fournies dans la section [Obtenez l'ARN du rôle de service \(CLI\)](#).

3. La politique gérée que vous utilisez dépend de la plateforme de calcul.
 - Si votre déploiement concerne une plate-forme informatique EC2/sur site :

Appelez la `attach-role-policy` commande pour donner au rôle de service nommé **CodeDeployServiceRole** les autorisations en fonction de la politique gérée IAM nommée **AWSCodeDeployRole**. Par exemple :

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole
```

- Si votre déploiement concerne une plate-forme de calcul AWS Lambda :

Appelez la `attach-role-policy` commande pour donner au rôle de service nommé **CodeDeployServiceRole** les autorisations en fonction de la politique gérée par IAM nommée **AWSCodeDeployRoleForLambda** ou **AWSCodeDeployRoleForLambdaLimited**. Par exemple :

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSCodeDeployRoleForLambda
```

- Si votre déploiement concerne une plateforme de calcul Amazon ECS :

Appelez la `attach-role-policy` commande pour donner au rôle de service nommé **CodeDeployServiceRole** les autorisations en fonction de la politique gérée par IAM nommée **AWSCodeDeployRoleForECS** ou **AWSCodeDeployRoleForECSLimited**. Par exemple :

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/AWSCodeDeployRoleForECS
```

Pour plus d'informations sur la création de rôles de service, consultez la section [Création d'un rôle pour un AWS service](#) dans le Guide de l'utilisateur IAM.

Obtenir l'ARN du rôle de service (console)

Pour utiliser la console IAM afin d'obtenir l'ARN du rôle de service :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le panneau de navigation, choisissez Rôles.
3. Dans la zone de texte Filtre, entrez **CodeDeployServiceRole**, puis appuyez sur Entrée.
4. Sélectionnez CodeDeployServiceRole.
5. Notez la valeur du champ Role ARN.

Obtenez l'ARN du rôle de service (CLI)

Pour utiliser le AWS CLI pour obtenir l'ARN du rôle de service, appelez la `get-role` commande en fonction du rôle de service nommé **CodeDeployServiceRole** :

```
aws iam get-role --role-name CodeDeployServiceRole --query "Role.Arn" --output text
```

La valeur retournée correspond au nom ARN du rôle de service.

Étape 3 : Limiter les autorisations de l' CodeDeploy utilisateur

Pour des raisons de sécurité, nous vous recommandons de limiter les autorisations de l'utilisateur administratif que vous avez créé [Étape 1 : Configuration](#) à celles requises pour créer et gérer des déploiements dans CodeDeploy.

Utilisez la série de procédures suivante pour limiter les autorisations de l'utilisateur CodeDeploy administratif.

Avant de commencer

- Assurez-vous d'avoir créé un utilisateur CodeDeploy administratif dans IAM Identity Center en [Étape 1 : Configuration](#) suivant les instructions de.

Pour créer un jeu d'autorisations

Vous attribuerez cet ensemble d'autorisations à l'utilisateur CodeDeploy administratif ultérieurement.

- Connectez-vous à la AWS IAM Identity Center console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/singlesignon/>.
- Dans le volet de navigation, choisissez Ensembles d'autorisations, puis choisissez Créer un ensemble d'autorisations.
- Choisissez Ensemble d'autorisations personnalisé.
- Choisissez Suivant.
- Choisissez Inline policy.
- Supprimez l'exemple de code.
- Ajoutez le code de politique suivant :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeDeployAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "autoscaling:*",
        "codedeploy:*",
        "ec2:*",
        "lambda:*",
        "ecs:*",
        "elasticloadbalancing:*",
        "iam:AddRoleToInstanceProfile",
        "iam:AttachRolePolicy",
```



```

        "iam:CreateInstanceProfile",
        "iam:CreateRole",
        "iam>DeleteInstanceProfile",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy",
        "iam:GetInstanceProfile",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfilesForRole",
        "iam:ListRolePolicies",
        "iam:ListRoles",
        "iam:PutRolePolicy",
        "iam:RemoveRoleFromInstanceProfile",
        "s3:*",
        "ssm:*"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeDeployRolePolicy",
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::111122223333:role/CodeDeployServiceRole"
  }
]
}

```

Dans cette politique, remplacez *arn:aws:iam::account-ID:role/CodeDeployServiceRole* par la valeur ARN du rôle de CodeDeploy service dans lequel vous l'avez créé [Étape 2 : créer un rôle de service pour CodeDeploy](#). Vous trouverez la valeur ARN sur la page de détails du rôle de service de la console IAM.

La politique précédente vous permet de déployer une application sur une plate-forme de calcul AWS Lambda, une plate-forme de calcul EC2/sur site et une plate-forme de calcul Amazon ECS.

Vous pouvez utiliser les CloudFormation modèles fournis dans cette documentation pour lancer des instances Amazon EC2 compatibles avec. CodeDeploy Pour utiliser des CloudFormation modèles pour créer des applications, des groupes de déploiement ou des configurations de

déploiement, vous devez fournir un CloudFormation accès aux AWS services et aux actions qui en CloudFormation dépendent en ajoutant l'`cloudformation:*autorisation` à la politique d'autorisation de l'utilisateur CodeDeploy administratif, comme suit :

JSON

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

8. Choisissez Suivant.
9. Dans Nom du jeu d'autorisations, entrez :

CodeDeployUserPermissionSet

10. Choisissez Suivant.
11. Sur la page Réviser et créer, passez en revue les informations et choisissez Créer.

Pour attribuer l'ensemble d'autorisations à l'utilisateur CodeDeploy administratif

1. Dans le volet de navigation, choisissez Comptes AWS, puis cochez la case à côté de Compte AWS celle à laquelle vous êtes actuellement connecté.
2. Cliquez sur le bouton Attribuer des utilisateurs ou des groupes.
3. Sélectionnez l'onglet Utilisateurs.
4. Cochez la case à côté de l'utilisateur CodeDeploy administratif.
5. Choisissez Suivant.
6. Cochez la case à côté de `CodeDeployUserPermissionSet`.
7. Choisissez Suivant.

8. Vérifiez les informations et choisissez Soumettre.

Vous avez maintenant attribué l'utilisateur CodeDeploy administratif et `CodeDeployUserPermissionSet` à votre Compte AWS, en les liant ensemble.

Pour vous déconnecter et vous reconnecter en tant qu'utilisateur CodeDeploy administratif

1. Avant de vous déconnecter, assurez-vous d'avoir l'URL du portail AWS d'accès ainsi que le nom d'utilisateur et le mot de passe à usage unique de l'utilisateur CodeDeploy administratif.

Note

Si vous ne disposez pas de ces informations, rendez-vous sur la page des informations de l'utilisateur CodeDeploy administratif dans IAM Identity Center, choisissez Réinitialiser le mot de passe, Générer un mot de passe à usage unique [...], et Réinitialisez à nouveau le mot de passe pour afficher les informations à l'écran.

2. Déconnectez-vous de AWS.
3. Collez l'URL du portail d' AWS accès dans la barre d'adresse de votre navigateur.
4. Connectez-vous en tant qu' CodeDeploy utilisateur administratif.

Une Compte AWSboîte apparaît à l'écran.

5. Choisissez Compte AWS, puis choisissez le nom auquel vous avez attribué l'utilisateur CodeDeploy administratif et le jeu d'autorisations. Compte AWS
6. À côté de `CodeDeployUserPermissionSet`, choisissez Console de gestion.

Le AWS Management Console apparaît. Vous êtes maintenant connecté en tant qu'utilisateur CodeDeploy administratif avec les autorisations limitées. Vous pouvez désormais effectuer des CodeDeploy opérations connexes, et uniquement des CodeDeploy opérations connexes, en tant que cet utilisateur.

Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2

Note

Si vous utilisez la plateforme de calcul Amazon ECS ou AWS Lambda, ignorez cette étape.

Vos instances Amazon EC2 ont besoin d'une autorisation pour accéder aux compartiments ou aux GitHub référentiels Amazon S3 dans lesquels les applications sont stockées. Pour lancer des instances Amazon EC2 compatibles avec CodeDeploy, vous devez créer un rôle IAM supplémentaire, un profil d'instance. Ces instructions vous montrent comment créer un profil d'instance IAM à associer à vos instances Amazon EC2. Ce rôle donne à l' CodeDeploy agent l'autorisation d'accéder aux compartiments ou aux GitHub référentiels Amazon S3 où sont stockées vos applications.

Vous pouvez créer un profil d'instance IAM à l'aide de la AWS CLI console IAM ou de l'IAM. APIs

Note

Vous pouvez attacher un profil d'instance IAM à une instance Amazon EC2 lorsque vous la lancez ou à une instance préalablement lancée. Pour plus d'informations, consultez la section [Profils d'instance](#).

Rubriques

- [Créez un profil d'instance IAM pour vos instances Amazon EC2 \(CLI\)](#)
- [Création d'un profil d'instance IAM pour vos instances Amazon EC2 \(console\)](#)

Créez un profil d'instance IAM pour vos instances Amazon EC2 (CLI)

Dans ces étapes, nous supposons que vous avez déjà suivi les instructions fournies dans les trois premières étapes de [Commencer avec CodeDeploy](#).

1. Sur votre machine de développement, créez un fichier texte nommé `CodeDeployDemo-EC2-Trust.json`. Collez le contenu suivant, qui permet à Amazon EC2 de travailler en votre nom :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ec2.cn-north-1.amazonaws.com",
          "ec2.cn-northwest-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Dans le même répertoire, créez un fichier texte nommé `CodeDeployDemo-EC2-Permissions.json`. Collez le contenu suivant :

```
{
```

```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": [
          "s3:Get*",
          "s3:List*"
        ],
        "Effect": "Allow",
        "Resource": "*"
      }
    ]
  }
}

```

Note

Nous vous recommandons de limiter cette politique aux seuls compartiments Amazon S3 auxquels vos instances Amazon EC2 doivent accéder. Assurez-vous de donner accès aux compartiments Amazon S3 qui contiennent l' CodeDeploy agent. Dans le cas contraire, une erreur peut se produire lors de l'installation ou de la mise à jour de l' CodeDeploy agent sur les instances. Pour accorder au profil d'instance IAM l'accès à certains compartiments du kit de CodeDeploy ressources uniquement dans Amazon S3, appliquez la politique suivante, mais supprimez les lignes relatives aux compartiments auxquels vous souhaitez empêcher l'accès :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",

```

```

    "arn:aws:s3::aws-codedeploy-eu-west-2/*",
    "arn:aws:s3::aws-codedeploy-eu-west-3/*",
    "arn:aws:s3::aws-codedeploy-eu-central-1/*",
    "arn:aws:s3::aws-codedeploy-eu-central-2/*",
    "arn:aws:s3::aws-codedeploy-eu-north-1/*",
    "arn:aws:s3::aws-codedeploy-eu-south-1/*",
    "arn:aws:s3::aws-codedeploy-eu-south-2/*",
    "arn:aws:s3::aws-codedeploy-il-central-1/*",
    "arn:aws:s3::aws-codedeploy-ap-east-1/*",
    "arn:aws:s3::aws-codedeploy-ap-northeast-1/*",
    "arn:aws:s3::aws-codedeploy-ap-northeast-2/*",
    "arn:aws:s3::aws-codedeploy-ap-northeast-3/*",
    "arn:aws:s3::aws-codedeploy-ap-southeast-1/*",
    "arn:aws:s3::aws-codedeploy-ap-southeast-2/*",
    "arn:aws:s3::aws-codedeploy-ap-southeast-3/*",
    "arn:aws:s3::aws-codedeploy-ap-southeast-4/*",
    "arn:aws:s3::aws-codedeploy-ap-south-1/*",
    "arn:aws:s3::aws-codedeploy-ap-south-2/*",
    "arn:aws:s3::aws-codedeploy-me-central-1/*",
    "arn:aws:s3::aws-codedeploy-me-south-1/*",
    "arn:aws:s3::aws-codedeploy-sa-east-1/*"
  ]
}
]
}

```

Note

Si vous souhaitez utiliser l'[autorisation IAM](#) ou les points de terminaison Amazon Virtual Private Cloud (VPC) avec CodeDeploy, vous devez ajouter des autorisations supplémentaires. Consultez [Utiliser CodeDeploy avec Amazon Virtual Private Cloud](#) pour plus d'informations.

3. À partir du même répertoire, appelez la `create-role` commande pour créer un rôle IAM nommé **CodeDeployDemo-EC2-Instance-Profile**, en fonction des informations contenues dans le premier fichier :

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

```
aws iam create-role --role-name CodeDeployDemo-EC2-Instance-Profile --assume-role-policy-document file://CodeDeployDemo-EC2-Trust.json
```

- À partir du même répertoire, appelez la commande `put-role-policy` pour donner au rôle nommé **CodeDeployDemo-EC2-Instance-Profile** les permissions basées sur les informations figurant dans le second fichier :

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

```
aws iam put-role-policy --role-name CodeDeployDemo-EC2-Instance-Profile --policy-name CodeDeployDemo-EC2-Permissions --policy-document file://CodeDeployDemo-EC2-Permissions.json
```

- Appellez le `attach-role-policy` pour accorder au rôle Amazon EC2 Systems Manager des autorisations afin que SSM puisse installer CodeDeploy l'agent. Cette politique n'est pas nécessaire si vous prévoyez d'installer l'agent depuis le compartiment public Amazon S3 à l'aide de la ligne de commande. En savoir plus sur [l'installation de l'agent CodeDeploy](#).

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore --role-name CodeDeployDemo-EC2-Instance-Profile
```

- Appellez la `create-instance-profile` command suivie de la `add-role-to-instance-profile` command pour créer un profil d'instance IAM nommé **CodeDeployDemo-EC2-Instance-Profile**. Le profil d'instance permet à Amazon EC2 de transmettre le rôle IAM nommé **CodeDeployDemo-EC2-Instance-Profile** à une instance Amazon EC2 lorsque celle-ci est lancée pour la première fois :


```
aws iam create-instance-profile --instance-profile-name CodeDeployDemo-EC2-Instance-Profile
aws iam add-role-to-instance-profile --instance-profile-name CodeDeployDemo-EC2-Instance-Profile --role-name CodeDeployDemo-EC2-Instance-Profile
```

Si vous avez besoin du nom du profil d'instance IAM, consultez [list-instance-profiles-for-role](#) dans la section IAM de la référence.AWS CLI

Vous avez maintenant créé un profil d'instance IAM à associer à vos instances Amazon EC2. Pour de plus amples informations sur les rôles IAM, consultez [Rôles IAM pour Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

Création d'un profil d'instance IAM pour vos instances Amazon EC2 (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le volet de navigation de la console IAM, sélectionnez Politiques, puis Create policy.
3. Sur la page Spécifier les autorisations, choisissez JSON.
4. Supprimez l'exemple JSON de code.
5. Collez le code suivant :

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

 Note

Nous vous recommandons de limiter cette politique aux seuls compartiments Amazon S3 auxquels vos instances Amazon EC2 doivent accéder. Assurez-vous de donner accès aux compartiments Amazon S3 qui contiennent l' CodeDeploy agent. Dans le cas contraire, une erreur peut se produire lors de l'installation ou de la mise à jour de l' CodeDeploy agent sur les instances. Pour accorder au profil d'instance IAM l'accès à certains compartiments du kit de CodeDeploy ressources uniquement dans Amazon S3, appliquez la politique suivante, mais supprimez les lignes relatives aux compartiments auxquels vous souhaitez empêcher l'accès :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
```

```
"arn:aws:s3::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3::aws-codedeploy-ap-southeast-3/*",
"arn:aws:s3::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3::aws-codedeploy-ap-south-1/*",
"arn:aws:s3::aws-codedeploy-ap-south-2/*",
"arn:aws:s3::aws-codedeploy-me-central-1/*",
"arn:aws:s3::aws-codedeploy-me-south-1/*",
"arn:aws:s3::aws-codedeploy-sa-east-1/*"
    ]
  }
]
}
```

 Note

Si vous souhaitez utiliser l'[autorisation IAM](#) ou les points de terminaison Amazon Virtual Private Cloud (VPC) avec CodeDeploy, vous devez ajouter des autorisations supplémentaires. Consultez [Utiliser CodeDeploy avec Amazon Virtual Private Cloud](#) pour plus d'informations.

6. Choisissez Suivant.
7. Sur la page Réviser et créer, dans la zone Nom de la politique, tapez **CodeDeployDemo-EC2-Permissions**.
8. (Facultatif) Dans le champ description, saisissez une description de la politique.
9. Sélectionnez Créer une politique.
10. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
11. Sous Cas d'utilisation, choisissez le cas d'utilisation EC2.
12. Choisissez Suivant.
13. Dans la liste des politiques, cochez la case à côté de la politique que vous venez de créer (CodeDeployDemo-EC2-Permissions). Si nécessaire, utilisez la zone de recherche pour trouver la politique.
14. Pour utiliser Systems Manager afin d'installer ou de configurer l' CodeDeploy agent, cochez la case à côté d'Amazon SSMManaged InstanceCore. Cette politique AWS gérée permet à une instance d'utiliser les fonctionnalités principales du service Systems Manager. Si nécessaire, utilisez la zone de recherche pour trouver la politique. Cette politique n'est pas nécessaire si

vous prévoyez d'installer l'agent depuis le compartiment public Amazon S3 à l'aide de la ligne de commande. En savoir plus sur [l'installation de l'agent CodeDeploy](#).

15. Choisissez Suivant.

16. Sur la page Nom, révision et création, dans Nom du rôle, entrez le nom du rôle de service (par exemple, **CodeDeployDemo-EC2-Instance-Profile**), puis choisissez Créer un rôle.

Vous pouvez également saisir une description de ce rôle de service dans Description du rôle.

Vous avez maintenant créé un profil d'instance IAM à associer à vos instances Amazon EC2. Pour de plus amples informations sur les rôles IAM, consultez [Rôles IAM pour Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

Intégrations de produits et de services avec CodeDeploy

Par défaut, CodeDeploy s'intègre à un certain nombre de AWS services et de produits et services partenaires. Les informations suivantes peuvent vous aider à configurer CodeDeploy afin de vous intégrer aux produits et services que vous utilisez.

- [Intégration avec d'autres AWS services](#)
- [Intégration avec les produits et services des partenaires](#)
- [Exemples d'intégration issus de la communauté](#)

Intégration avec d'autres AWS services

CodeDeploy est intégré aux AWS services suivants :

Amazon CloudWatch

[Amazon CloudWatch](#) est un service de surveillance des ressources du AWS cloud et des applications que vous utilisez AWS. Vous pouvez utiliser Amazon CloudWatch pour collecter et suivre les métriques, collecter et surveiller les fichiers journaux et définir des alarmes. CodeDeploy prend en charge les CloudWatch outils suivants :

- CloudWatch des alarmes pour surveiller vos déploiements et les arrêter lorsque les mesures de surveillance que vous avez spécifiées dépassent ou tombent en dessous des seuils que vous spécifiez dans une règle CloudWatch d'alarme. Pour utiliser la surveillance des alarmes, vous devez d'abord configurer une alarme CloudWatch, puis l'ajouter CodeDeploy à l'application ou au groupe de déploiement dans lequel les déploiements doivent s'arrêter lorsque l'alarme est activée.

En savoir plus :

- [Création de CloudWatch journaux et d'alarmes](#)
- Amazon CloudWatch Events pour détecter et réagir aux modifications de l'état d'une instance ou d'un déploiement dans le cadre de vos CodeDeploy opérations. Ensuite, en fonction des règles que vous créez, CloudWatch Events invoque une ou plusieurs actions cibles lorsqu'un déploiement ou une instance entre dans l'état que vous spécifiez dans une règle.

En savoir plus :

- [Surveillance des déploiements avec Amazon CloudWatch Events](#)
- Amazon CloudWatch Logs pour surveiller les trois types de journaux créés par l'agent CodeDeploy sans avoir à se connecter aux instances un par un.

En savoir plus :

- [Afficher les CodeDeploy journaux dans la console CloudWatch Logs](#)

Amazon EC2 Auto Scaling

CodeDeploy prend en charge [Amazon EC2 Auto Scaling](#). Ce AWS service peut lancer automatiquement des instances Amazon EC2 en fonction de critères que vous spécifiez, par exemple :

- Limites dépassées pour l'utilisation d'UC spécifiée.
- Lectures/écritures sur disque.
- Trafic réseau entrant ou sortant sur un intervalle de temps spécifié.

Vous pouvez étendre un groupe d'instances Amazon EC2 chaque fois que vous en avez besoin, puis les utiliser CodeDeploy pour y déployer automatiquement des révisions d'applications. Amazon EC2 Auto Scaling met fin à ces instances Amazon EC2 lorsqu'elles ne sont plus nécessaires.

En savoir plus :

- [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#)
- [Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling](#)
- [Sous le capot : CodeDeploy et intégration avec Auto Scaling](#)

Amazon Elastic Container Service

Vous pouvez l'utiliser CodeDeploy pour déployer une application conteneurisée Amazon ECS sous forme d'ensemble de tâches. CodeDeploy effectue un blue/green déploiement en installant une version mise à jour de l'application en tant que nouvel ensemble de tâches de remplacement. CodeDeploy redirige le trafic de production de l'ensemble de tâches d'application d'origine vers le jeu de tâches de remplacement. L'ensemble de tâches d'origine est arrêté après un déploiement réussi. Pour plus d'informations sur Amazon ECS, consultez [Amazon Elastic Container Service](#).

Vous pouvez gérer la manière dont le trafic est transféré vers l'ensemble de tâches mis à jour au cours d'un déploiement en choisissant une configuration Canary, Linear ou une all-at-once configuration. Pour plus d'informations sur les déploiements Amazon ECS, consultez la section [Déploiements sur une plateforme de calcul Amazon ECS](#).

AWS CloudTrail

CodeDeploy est intégré à [AWS CloudTrail](#). Ce service capture les appels d'API effectués par ou au nom de votre AWS compte et envoie les fichiers journaux CodeDeploy dans un compartiment Amazon S3 que vous spécifiez. CloudTrail capture les appels d'API à partir de la CodeDeploy console AWS CLI, à partir de CodeDeploy commandes via ou CodeDeploy APIs directement. Grâce aux informations collectées par CloudTrail, vous pouvez déterminer :

- Quelle demande a été faite à CodeDeploy.
- l'adresse IP source à partir de laquelle la requête a été effectuée ;
- la personne ayant formulé la requête ;
- le moment où la requête a été formulée.

En savoir plus :

- [Monitoring Deployments](#)

AWS Cloud9

[AWS Cloud9](#) est un environnement de développement intégré (IDE) en ligne basé sur le cloud que vous pouvez utiliser pour écrire, exécuter, déboguer et déployer du code à l'aide d'un simple navigateur depuis une machine connectée à Internet. AWS Cloud9 inclut un éditeur de code, un débogueur, un terminal et des outils essentiels, tels que Git et AWS CLI Git.

- Vous pouvez utiliser l' AWS Cloud9 IDE pour exécuter, déboguer et créer du code qui se trouve dans un GitHub référentiel. Il est possible d'afficher, de modifier et d'enregistrer le code dans la fenêtre Environment (Environnement) et les onglets de l'éditeur dans l'IDE. Lorsque vous êtes prêt, vous pouvez utiliser Git dans la session du AWS Cloud9 terminal pour transférer les modifications de code dans votre GitHub dépôt, puis AWS CodeDeploy pour déployer vos mises à jour. Pour plus d'informations sur l'utilisation AWS Cloud9 avec GitHub, consultez [GitHub l'exemple de AWS Cloud9](#).
- Vous pouvez utiliser l' AWS Cloud9 IDE pour mettre à jour une AWS Lambda fonction. Vous pouvez ensuite l'utiliser AWS CodeDeploy pour créer un déploiement qui déplace le trafic vers la nouvelle version de votre AWS Lambda fonction. Pour plus d'informations, consultez la section [Utilisation des AWS Lambda fonctions dans l'environnement de développement AWS Cloud9 intégré \(IDE\)](#).

Pour plus d'informations AWS Cloud9, voir [Qu'est-ce que c'est AWS Cloud9](#) et [comment démarrer avec AWS Cloud9](#).

AWS CodePipeline

[AWS CodePipeline](#) est un service de diffusion en continu qui permet de modéliser, visualiser et automatiser les étapes nécessaires à la publication de vos logiciels dans un processus de diffusion en continu. Vous pouvez utiliser AWS CodePipeline pour définir votre propre processus de publication afin que le service crée, teste et déploie votre code chaque fois qu'une modification de code intervient. Par exemple, vous pourriez disposer de trois groupes de déploiement pour une application : bêta, gamma et production. Vous pouvez configurer un pipeline de telle sorte que chaque fois qu'une modification intervient dans votre code source, les mises à jour sont déployées individuellement dans chaque groupe de déploiement.

Vous pouvez configurer AWS CodePipeline pour utiliser CodeDeploy pour déployer :

- Code pour les instances Amazon EC2, les instances sur site ou les deux.
- Versions de la fonction AWS Lambda sans serveur.

Vous pouvez créer l'application CodeDeploy, le déploiement et le groupe de déploiement à utiliser dans une action de déploiement à une étape, soit avant de créer le pipeline, soit dans l'assistant de création de pipeline.

En savoir plus :

- [AWS pour le guide de DevOps démarrage](#) — Découvrez comment utiliser CodePipeline

pour fournir et déployer du code source en continu dans des CodeCommit référentiels vers des instances Amazon EC2. CodeDeploy

- [Procédure pas à pas simple du pipeline \(Amazon S3 Bucket\)](#)
- [Procédure pas à pas simple du pipeline \(CodeCommit référentiel\)](#)
- [Tutoriel en quatre étapes sur le pipeline](#)

AWS Modèle d'application sans serveur

AWS Le modèle d'application sans serveur (AWS SAM) est un modèle permettant de définir des applications sans serveur. Il propose également CloudFormation un moyen simplifié de définir AWS Lambda les fonctions, Amazon API Gateway APIs et les tables Amazon DynamoDB requises par une application sans serveur. Si vous utilisez déjà AWS SAM, vous pouvez ajouter des préférences de déploiement à utiliser pour commencer CodeDeploy à gérer la manière dont le trafic est transféré lors du déploiement d'une application AWS Lambda.

Pour plus d'informations, consultez le [modèle d'application AWS sans serveur](#).

Elastic Load Balancing

CodeDeploy prend en [charge Elastic Load Balancing](#), un service qui distribue le trafic applicatif entrant sur plusieurs instances Amazon EC2.

Pour CodeDeploy les déploiements, les équilibres de charge empêchent également le trafic d'être acheminé vers des instances lorsqu'elles ne sont pas prêtes, sont en cours de déploiement ou ne sont plus nécessaires dans le cadre d'un environnement.

En savoir plus :

- [Integrating CodeDeploy with Elastic Load Balancing](#)

Rubriques

- [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#)
- [Intégration CodeDeploy à Elastic Load Balancing](#)

Intégration CodeDeploy à Amazon EC2 Auto Scaling

CodeDeploy prend en charge Amazon EC2 Auto Scaling, AWS un service qui lance les instances Amazon EC2 automatiquement en fonction des conditions que vous définissez. Ces conditions peuvent inclure le dépassement des limites dans un intervalle de temps spécifié pour l'utilisation du processeur, les lectures ou écritures sur le disque, ou le trafic réseau entrant ou sortant. Amazon EC2 Auto Scaling met fin aux instances lorsqu'elles ne sont plus nécessaires. Pour plus d'informations, consultez [Qu'est-ce que Amazon EC2 Auto Scaling ?](#) dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Lorsque de nouvelles instances Amazon EC2 sont lancées dans le cadre d'un groupe Amazon EC2 Auto Scaling, CodeDeploy vous pouvez déployer automatiquement vos révisions sur les nouvelles instances. Vous pouvez également coordonner les déploiements CodeDeploy avec des instances Amazon EC2 Auto Scaling enregistrées auprès des équilibres de charge Elastic Load Balancing. Pour plus d'informations, consultez [Integrating CodeDeploy with Elastic Load Balancing](#)

et [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#).

Note

Vous pouvez rencontrer des problèmes si vous associez plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling. Si un déploiement échoue, par exemple, l'instance va commencer à s'arrêter, mais les autres déploiements en cours d'exécution peuvent mettre une heure avant d'expirer. Pour plus d'informations, consultez [Évitez d'associer plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling Under the hood : CodeDeploy et Amazon EC2 Auto Scaling integration](#).

Rubriques

- [Déploiement CodeDeploy d'applications dans des groupes Amazon EC2 Auto Scaling](#)
- [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#)
- [Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy](#)
- [Utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling](#)

Déploiement CodeDeploy d'applications dans des groupes Amazon EC2 Auto Scaling

Pour déployer une révision d' CodeDeploy application dans un groupe Amazon EC2 Auto Scaling :

1. Créez ou localisez un profil d'instance IAM qui permet au groupe Amazon EC2 Auto Scaling de travailler avec Amazon S3. Pour de plus amples informations, veuillez consulter [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).

Note

Vous pouvez également l'utiliser CodeDeploy pour déployer des révisions depuis des GitHub référentiels vers des groupes Amazon EC2 Auto Scaling. Bien que les instances Amazon EC2 nécessitent toujours un profil d'instance IAM, le profil n'a pas besoin d'autorisations supplémentaires pour être déployé à partir d'un référentiel. GitHub

2. Créez ou utilisez un groupe Amazon EC2 Auto Scaling, en spécifiant le profil d'instance IAM dans votre configuration ou votre modèle de lancement. Pour plus d'informations, consultez le [rôle IAM pour les applications qui s'exécutent sur des instances Amazon EC2](#).

3. Créez ou localisez un rôle de service qui permet CodeDeploy de créer un groupe de déploiement contenant le groupe Amazon EC2 Auto Scaling.
4. Créez un groupe de déploiement avec CodeDeploy, en spécifiant le nom du groupe Amazon EC2 Auto Scaling, le rôle du service et quelques autres options. Pour plus d'informations, consultez [Création d'un groupe de déploiement pour un déploiement sur place \(console\)](#) ou [Création d'un groupe de déploiement pour un déploiement sur place \(console\)](#).
5. CodeDeploy À utiliser pour déployer votre révision dans le groupe de déploiement qui contient le groupe Amazon EC2 Auto Scaling.

Pour de plus amples informations, veuillez consulter [Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling](#).

Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling

Un déploiement de fin est un type de CodeDeploy déploiement qui est activé automatiquement lorsqu'un [événement de scale-in Auto Scaling](#) se produit. CodeDeploy effectue le déploiement de terminaison juste avant que le service Auto Scaling ne mette fin à l'instance. Lors d'un déploiement final, CodeDeploy ne déploie rien. Au lieu de cela, il génère des événements du cycle de vie, que vous pouvez associer à vos propres scripts pour activer une fonctionnalité d'arrêt personnalisée. Par exemple, vous pouvez associer l'événement du `ApplicationStop` cycle de vie à un script qui arrête correctement votre application avant que l'instance ne soit arrêtée.

Pour obtenir la liste des événements du cycle de vie CodeDeploy générés lors de l'arrêt d'un déploiement, consultez [Disponibilité du carnet d'événements du cycle de vie](#).

Si le déploiement de la terminaison échoue pour une raison quelconque, CodeDeploy cela permettra à la résiliation de l'instance de se poursuivre. Cela signifie que l'instance sera arrêtée même si elle CodeDeploy n'a pas exécuté l'ensemble complet (ou aucun) des événements du cycle de vie jusqu'à son terme.

Si vous n'activez pas les déploiements de résiliation, le service Auto Scaling mettra tout de même fin aux instances Amazon EC2 lorsqu'un événement de scale-in se produit, CodeDeploy mais ne générera pas d'événements liés au cycle de vie.

 Note

Que vous activiez ou non les déploiements de résiliation, si le service Auto Scaling met fin à une instance Amazon EC2 alors qu'un CodeDeploy déploiement est en cours, une course peut se produire entre les événements du cycle de vie générés par Auto Scaling et les services. CodeDeploy Par exemple, l'événement `Terminating` du cycle de vie (généré par le service Auto Scaling) peut remplacer l'`ApplicationStart` événement (généré par le CodeDeploy déploiement). Dans ce scénario, il est possible que vous rencontriez un échec lors de la fermeture ou du déploiement de l'instance Amazon EC2.

Pour permettre à CodeDeploy d'effectuer des déploiements de terminaison

- Cochez la case Ajouter un crochet de terminaison aux groupes Auto Scaling lors de la création ou de la mise à jour de votre groupe de déploiement. Pour obtenir des instructions [Création d'un groupe de déploiement pour un déploiement sur place \(console\)](#), voir ou [Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site \(console\)](#).

L'activation de cette case CodeDeploy à cocher entraîne l'installation d'un [hook de cycle de vie Auto Scaling](#) dans les groupes Auto Scaling que vous spécifiez lorsque vous créez ou mettez à jour votre groupe de CodeDeploy déploiement. Ce crochet est appelé crochet de terminaison et permet les déploiements de terminaison.

Une fois le crochet de terminaison installé, un événement scale-in (terminaison) se déroule comme suit :

1. Le service Auto Scaling (ou simplement Auto Scaling) détermine qu'un événement de scale-in doit se produire et contacte le service EC2 pour mettre fin à une instance EC2.
2. Le service EC2 commence à mettre fin à l'instance EC2. L'instance passe dans l'`Terminating` état, puis dans l'`Terminating:Wait` état.
3. Pendant `Terminating:Wait` ce temps, Auto Scaling exécute tous les hooks du cycle de vie attachés au groupe Auto Scaling, y compris le crochet de terminaison installé par CodeDeploy.
4. Le hook de terminaison envoie une notification à la [file d'attente Amazon SQS](#) interrogée par CodeDeploy.
5. Dès réception de la notification, CodeDeploy analyse le message, effectue une certaine validation et effectue un [déploiement de terminaison](#).

6. Pendant que le déploiement de la terminaison est en cours, CodeDeploy envoie des pulsations toutes les cinq minutes à Auto Scaling pour l'informer que l'instance est toujours en cours d'élaboration.
7. Jusqu'à présent, l'instance EC2 est toujours dans `Terminating:Wait` cet état (ou peut-être dans cet `Warmup:Pending:Wait` état, si vous avez activé les [warm pools de groupe Auto Scaling](#)).
8. Lorsque le déploiement est terminé, CodeDeploy indique à Auto Scaling CONTINUE le processus d'arrêt d'EC2, que le déploiement ait réussi ou échoué.

Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy

Lorsque vous créez ou mettez à jour un groupe de CodeDeploy déploiement pour inclure un groupe Auto Scaling, vous CodeDeploy accédez au groupe Auto Scaling à l'aide du rôle de CodeDeploy service, puis installez les [hooks de cycle de vie Auto Scaling](#) dans vos groupes Auto Scaling.

Note

Les hooks du cycle de vie Auto Scaling sont différents des événements du cycle de vie (également appelés crochets d'événements du cycle [AppSpec section « crochets »](#) de vie) générés CodeDeploy et décrits dans ce guide.

Les hooks du cycle de vie Auto Scaling qui s' CodeDeploy installent sont les suivants :

- Un hook de lancement : ce hook indique CodeDeploy qu'un [événement de scale-out Auto Scaling](#) est en cours et qu'il CodeDeploy doit démarrer un déploiement de lancement.

Lors d'un déploiement de lancement, CodeDeploy :

- Déploie une révision de votre application sur l'instance évolutive.
- Génère des événements du cycle de vie pour indiquer la progression du déploiement. Vous pouvez associer ces événements du cycle de vie à vos propres scripts pour activer des fonctionnalités de démarrage personnalisées. Pour plus d'informations, consultez le tableau dans [Disponibilité du carnet d'événements du cycle de vie](#).

Le hook de lancement et le déploiement de lancement associé sont toujours activés et ne peuvent pas être désactivés.

- Un hook de terminaison : ce hook facultatif indique CodeDeploy qu'un [événement de scale-in Auto Scaling](#) est en cours et qu'il CodeDeploy doit démarrer un déploiement de terminaison.

Lors d'un déploiement d'arrêt, CodeDeploy génère des événements du cycle de vie pour indiquer la progression de l'arrêt de l'instance. Pour de plus amples informations, veuillez consulter [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#).

Rubriques

- [Après avoir CodeDeploy installé les hooks Lifecycle, comment sont-ils utilisés ?](#)
- [Comment sont CodeDeploy nommés les groupes Amazon EC2 Auto Scaling](#)
- [Ordre d'exécution des événements d'accroche du cycle de vie personnalisés](#)
- [Événements de scale-out lors d'un déploiement](#)
- [Événements d'évolutivité lors d'un déploiement](#)
- [Ordre des événements dans les scripts AWS CloudFormation cfn-init](#)

Après avoir CodeDeploy installé les hooks Lifecycle, comment sont-ils utilisés ?

Une fois les hooks du cycle de vie de lancement et de fin installés, ils sont utilisés CodeDeploy lors des événements de scale-out et de scale-in du groupe Auto Scaling, respectivement.

Un événement de scale-out (lancement) se déroule comme suit :

1. Le service Auto Scaling (ou simplement Auto Scaling) détermine qu'un événement de scale-out doit se produire et contacte le service EC2 pour lancer une nouvelle instance EC2.
2. Le service EC2 lance une nouvelle instance EC2. L'instance passe dans l'`Pending` état, puis dans l'`Pending:Wait` état.
3. Pendant `Pending:Wait` ce temps, Auto Scaling exécute tous les hooks du cycle de vie attachés au groupe Auto Scaling, y compris le hook de lancement installé par CodeDeploy.
4. Le hook de lancement envoie une notification à la [file d'attente Amazon SQS interrogée](#) par CodeDeploy
5. Dès réception de la notification, CodeDeploy analyse le message, effectue une validation et lance un [déploiement de lancement](#).
6. Pendant le déploiement de lancement, CodeDeploy envoie des pulsations toutes les cinq minutes à Auto Scaling pour l'informer que l'instance est toujours en cours d'élaboration.
7. Jusqu'à présent, l'instance EC2 est toujours dans `Pending:Wait` cet état.
8. Lorsque le déploiement est terminé, CodeDeploy indique à Auto Scaling l'un `CONTINUE` ou `ABANDON` l'autre processus de lancement d'EC2, selon que le déploiement a réussi ou échoué.

- Si CodeDeploy cela est indiqué `CONTINUE`, Auto Scaling poursuit le processus de lancement, soit en attendant que les autres hooks soient terminés, soit en plaçant l'instance dans l'`InService` état « `Pending:Proceed` et puis ».
- Si CodeDeploy cela est indiqué `ABANDON`, Auto Scaling met fin à l'instance EC2 et redémarre la procédure de lancement si nécessaire pour atteindre le nombre d'instances souhaité, tel que défini dans le paramètre Auto Scaling `Desired Capacity`.

Un événement de scale-in (résiliation) se déroule comme suit :

Consultez [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#).

Comment sont CodeDeploy nommés les groupes Amazon EC2 Auto Scaling

Lors blue/green des déploiements sur une plate-forme informatique EC2/sur site, deux options s'offrent à vous pour ajouter des instances à votre environnement de remplacement (écologique) :

- Utilisez des instances qui existent déjà ou créez-en manuellement.
- Utilisez les paramètres d'un groupe Amazon EC2 Auto Scaling que vous spécifiez pour définir et créer des instances dans un nouveau groupe Amazon EC2 Auto Scaling.

Si vous choisissez la deuxième option, CodeDeploy provisionne un nouveau groupe Amazon EC2 Auto Scaling pour vous. Il utilise la convention suivante pour nommer le groupe :

```
CodeDeploy_deployment_group_name_deployment_id
```

Par exemple, si un déploiement avec ID 10 déploie un groupe de déploiement nommé `alpha-deployments`, le groupe Amazon EC2 Auto Scaling provisionné est nommé `CodeDeploy_alpha-deployments_10`. Pour plus d'informations, consultez [Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site \(console\)](#) et [GreenFleetProvisioningOption](#).

Ordre d'exécution des événements d'accroche du cycle de vie personnalisés

Vous pouvez ajouter vos propres hooks de cycle de vie aux groupes Amazon EC2 Auto Scaling vers CodeDeploy lesquels sont déployés. Toutefois, l'ordre dans lequel ces événements d'accroche du cycle de vie personnalisés sont exécutés ne peut pas être prédéterminé par rapport aux événements du cycle de vie de déploiement CodeDeploy par défaut. Par exemple, si vous ajoutez un hook de cycle de vie personnalisé nommé `ReadyForSoftwareInstall` dans un groupe Amazon EC2 Auto

Scaling, vous ne pouvez pas savoir à l'avance s'il sera exécuté avant le premier ou après le dernier CodeDeploy événement du cycle de vie de déploiement par défaut.

Pour savoir comment ajouter des hooks de cycle de vie personnalisés à un groupe Amazon EC2 Auto Scaling, [consultez la section Ajout de crochets de cycle](#) de vie dans le guide de l'utilisateur d'Amazon EC2 Auto Scaling.

Événements de scale-out lors d'un déploiement

Si un événement Auto Scale-out survient alors qu'un déploiement est en cours, les nouvelles instances seront mises à jour avec la version de l'application précédemment déployée, et non avec la version la plus récente de l'application. Si le déploiement aboutit, les anciennes instances et les nouvelles instances évolutives hébergeront différentes révisions de l'application. Pour mettre à jour les instances dotées de l'ancienne révision, lance CodeDeploy automatiquement un déploiement de suivi (immédiatement après la première) afin de mettre à jour les instances obsolètes. Si vous souhaitez modifier ce comportement par défaut afin que les instances EC2 obsolètes soient conservées dans l'ancienne version, consultez [Automatic updates to outdated instances](#).

Si vous souhaitez suspendre les processus de scale-out d'Amazon EC2 Auto Scaling pendant les déploiements, vous pouvez le faire via un paramètre `common_functions.sh` du script utilisé pour l'équilibrage de charge avec. CodeDeploy Si `HANDLE_PROCS=true` les événements Auto Scaling suivants sont automatiquement suspendus pendant le processus de déploiement :

- AZRebalance
- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

Important

Seul le CodeDeployDefault. OneAtATime la configuration de déploiement prend en charge cette fonctionnalité.

Pour plus d'informations sur l'utilisation `HANDLE_PROCS=true` afin d'éviter les problèmes de déploiement lors de l'utilisation d'Amazon EC2 Auto Scaling, [consultez la section Avis important concernant la AutoScaling gestion des](#) processus [aws-codedeploy-samples](#) en GitHub cours.

Événements d'évolutivité lors d'un déploiement

Si un groupe Auto Scaling commence à prendre de l'ampleur alors qu'un CodeDeploy déploiement est en cours sur ce groupe Auto Scaling, une situation de course peut survenir entre le processus d'arrêt (y compris les événements du cycle de vie du déploiement de fin) et les autres événements CodeDeploy du cycle de vie sur l'instance de CodeDeploy résiliation. Le déploiement sur cette instance spécifique peut échouer si l'instance est arrêtée avant la fin de tous les événements CodeDeploy du cycle de vie. En outre, le CodeDeploy déploiement global peut échouer ou non, selon la façon dont vous avez défini le paramètre Minimum healthy hosts dans votre configuration de déploiement.

Ordre des événements dans les scripts AWS CloudFormation cfn-init

Si vous utilisez `cfn-init` (ou `cloud-init`) pour exécuter des scripts sur des instances basées sur Linux nouvellement allouées, vos déploiements peuvent échouer à moins que vous ne contrôliez de manière stricte l'ordre des événements qui se produisent au démarrage de l'instance.

L'ordre doit être le suivant :

1. L'instance nouvellement allouée se lance.
2. Tous les scripts d'amorçage `cfn-init` s'exécutent jusqu'à la fin.
3. L' CodeDeploy agent démarre.
4. La dernière révision d'application est déployée sur l'instance.

Si l'ordre des événements n'est pas soigneusement contrôlé, l' CodeDeploy agent peut démarrer un déploiement avant la fin de l'exécution de tous les scripts.

Pour contrôler l'ordre des événements, utilisez l'une de ces bonnes pratiques :

- Installez l' CodeDeploy agent par le biais d'un `cfn-init` script, en le plaçant après tous les autres scripts.
- Incluez l' CodeDeploy agent dans une AMI personnalisée et utilisez un `cfn-init` script pour le démarrer, en le plaçant après tous les autres scripts.

Pour plus d'informations sur l'utilisation `cfn-init`, consultez [cfn-init](#) dans le guide de l'AWS CloudFormation utilisateur.

Utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling

Vous avez deux options pour spécifier l'AMI de base à utiliser lorsque de nouvelles instances Amazon EC2 sont lancées dans un groupe Amazon EC2 Auto Scaling :

- Vous pouvez spécifier une AMI personnalisée de base sur laquelle l' CodeDeploy agent est déjà installé. L'agent étant déjà installé, cette option lance les nouvelles instances Amazon EC2 plus rapidement que l'autre option. Toutefois, cette option augmente le risque d'échec des déploiements initiaux d'instances Amazon EC2, en particulier si CodeDeploy l'agent n'est pas à jour. Si vous choisissez cette option, nous vous recommandons de mettre régulièrement à jour l' CodeDeploy agent dans votre AMI personnalisée de base.
- Vous pouvez spécifier une AMI de base sur laquelle l' CodeDeploy agent n'est pas installé et l'installer lorsque chaque nouvelle instance est lancée dans un groupe Amazon EC2 Auto Scaling. Bien que cette option lance les nouvelles instances Amazon EC2 plus lentement que l'autre option, elle augmente les chances de réussite des déploiements initiaux d'instances. Cette option utilise la version la plus récente de l'agent CodeDeploy.

Intégration CodeDeploy à Elastic Load Balancing

Lors CodeDeploy des déploiements, un équilibreur de charge empêche le trafic Internet d'être acheminé vers des instances lorsqu'elles ne sont pas prêtes, sont en cours de déploiement ou ne sont plus nécessaires dans le cadre d'un environnement. Le rôle exact joué par l'équilibreur de charge dépend toutefois de son utilisation dans le cadre d'un blue/green déploiement ou d'un déploiement sur place.

Note

L'utilisation d'équilibreurs de charge Elastic Load Balancing est obligatoire dans les déploiements et facultative dans blue/green les déploiements sur place.

Types d'équilibreurs de charge Elastic Load Balancing

Elastic Load Balancing propose trois types d'équilibreurs de charge qui peuvent être utilisés dans les CodeDeploy déploiements : les équilibreurs de charge classiques, les équilibreurs de charge d'application et les équilibreurs de charge réseau.

Classic Load Balancer

Les itinéraires et les équilibres de charge se situent soit au niveau de la couche de transport (TCP/SSL) or the application layer (HTTP/HTTPS). Il prend en charge un VPC.

Note

Les équilibreurs de charge classiques ne sont pas pris en charge avec les déploiements Amazon ECS.

Application Load Balancer

Achemine et équilibre les charges au niveau de la couche d'application (HTTP/HTTPS) et prend en charge le routage basé sur le chemin d'accès. Il peut acheminer les requêtes vers des ports sur chaque instance EC2 ou de conteneur dans votre Virtual Private Cloud (VPC).

Note

Les groupes cibles d'Application Load Balancer doivent avoir un type de cible pour les déploiements sur des instances EC2 et instance pour IP les déploiements Fargate. Pour plus d'informations, consultez la section [Type de cible](#).

Network Load Balancer

Routes et équilibres de charge au niveau de la couche transport (couche TCP/UDP 4) sur la base des informations d'adresse extraites de l'en-tête du paquet TCP, et non du contenu du paquet. Un équilibreur de charge Network Load Balancer peut traiter les pics de trafic, conserver l'adresse IP source du client et utiliser une adresse IP fixe pendant la durée de vie de l'équilibreur de charge.

Pour en savoir plus sur les équilibreurs de charge Elastic Load Balancing, consultez les rubriques suivantes :

- [Qu'est-ce qu'Elastic Load Balancing ?](#)
- [Qu'est-ce qu'un équilibreur de charge Classic Load Balancer ?](#)
- [Qu'est-ce qu'un équilibreur de charge Application Load Balancer ?](#)
- [Qu'est-ce qu'un équilibreur de charge Network Load Balancer ?](#)

Déploiements bleu/vert

Le réacheminement du trafic d'instance via un équilibreur de charge Elastic Load Balancing est fondamental pour CodeDeploy blue/green les déploiements.

Lors d'un blue/green déploiement, l'équilibreur de charge permet d'acheminer le trafic vers les nouvelles instances d'un groupe de déploiement dans lequel la dernière révision de l'application a été déployée (l'environnement de remplacement), conformément aux règles que vous spécifiez, puis bloque le trafic provenant des anciennes instances sur lesquelles la révision précédente de l'application était exécutée (l'environnement d'origine).

Une fois que les instances d'un environnement de remplacement sont enregistrées auprès d'un ou de plusieurs équilibreurs de charge, les instances de l'environnement d'origine sont désenregistrées et, si vous le souhaitez, résiliées.

Pour un blue/green déploiement, vous pouvez spécifier un ou plusieurs groupes cibles Classic Load Balancer, Application Load Balancer ou Network Load Balancer dans votre groupe de déploiement. Vous utilisez la CodeDeploy console ou AWS CLI pour ajouter les équilibreurs de charge à un groupe de déploiement.

Pour plus d'informations sur les équilibreurs de charge dans blue/green les déploiements, consultez les rubriques suivantes :

- [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#)
- [Création d'une application pour un blue/green déploiement \(console\)](#)
- [Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site \(console\)](#)

Déploiements sur place

Lors d'un déploiement sur place, un équilibreur de charge empêche le trafic Internet d'être acheminé vers une instance alors qu'il est déployé sur cette dernière, et ensuite il met de nouveau l'instance à disposition du trafic après la fin du déploiement sur cette dernière.

Si un équilibreur de charge n'est pas utilisé pendant un déploiement sur place, le trafic Internet peut continuer à être dirigé vers une instance au cours du processus de déploiement. Vos clients risquent donc de rencontrer des applications web rompues, incomplètes ou obsolètes. Lorsque vous utilisez un équilibreur de charge Elastic Load Balancing dans le cadre d'un déploiement sur place,

les instances d'un groupe de déploiement sont désenregistrées de l'équilibreur de charge, mises à jour avec la dernière révision de l'application, puis réenregistrées auprès de l'équilibreur de charge dans le cadre du même groupe de déploiement une fois le déploiement réussi. CodeDeploy attendra jusqu'à 1 heure pour que l'instance soit saine derrière l'équilibreur de charge. Si l'instance n'est pas marquée comme saine par l'équilibreur de charge pendant la période d'attente, CodeDeploy elle passe à l'instance suivante ou échoue, selon la configuration du déploiement.

Pour un déploiement sur place, vous pouvez spécifier un ou plusieurs Classic Load Balancers, des groupes cibles Application Load Balancer ou des groupes cibles Network Load Balancer. Vous pouvez spécifier les équilibreurs de charge dans le cadre de la configuration du groupe de déploiement, ou vous pouvez utiliser un script fourni par CodeDeploy pour implémenter les équilibreurs de charge.

Spécifiez un équilibreur de charge de déploiement sur place à l'aide d'un groupe de déploiement

Pour ajouter des équilibreurs de charge à un groupe de déploiement, vous utilisez la CodeDeploy console ou AWS CLI. Pour plus d'informations sur la spécification d'un équilibreur de charge dans un groupe de déploiement pour les déploiements sur place, consultez les rubriques suivantes :

- [Création d'une application pour un déploiement sur place \(console\)](#)
- [Création d'un groupe de déploiement pour un déploiement sur place \(console\)](#)
- [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#)

Spécifiez un équilibreur de charge de déploiement sur place à l'aide d'un script

Utilisez les étapes de la procédure suivante pour utiliser les scripts de cycle de vie du déploiement et configurer l'équilibrage de charge pour les déploiements sur place.

Note

Vous devez utiliser le CodeDeployDefault.OneAtATime configuration de déploiement uniquement lorsque vous utilisez un script pour configurer un équilibreur de charge pour un déploiement sur place. Les exécutions simultanées ne sont pas prises en charge, et le CodeDeployDefault.OneAtATime ce paramètre garantit une exécution en série des scripts. Pour plus d'informations sur les configurations de déploiement, consultez [Utilisation des configurations de déploiement dans CodeDeploy](#).

Dans le référentiel CodeDeploy Samples GitHub, nous fournissons des instructions et des exemples que vous pouvez adapter pour utiliser les équilibreurs de charge CodeDeploy Elastic Load Balancing. Ces référentiels incluent trois exemples de scripts (`register_with_elb.sh`, `deregister_from_elb.sh`, `etcommon_functions.sh`) qui fournissent tout le code dont vous avez besoin pour démarrer. Il vous suffit de modifier les espaces réservés dans ces trois scripts, puis de faire référence à ces scripts à partir de votre fichier `appspec.yml`.

Pour configurer des déploiements sur place CodeDeploy avec des instances Amazon EC2 enregistrées auprès des équilibreurs de charge Elastic Load Balancing, procédez comme suit :

1. Téléchargez les exemples pour le type d'équilibreur de charge que vous voulez utiliser pour un déploiement sur place :
 - [Équilibreur de charge classique](#)
 - [Application Load Balancer ou Network Load Balancer \(le même script peut être utilisé pour les deux types\)](#)
2. Assurez-vous que chacune de vos instances Amazon EC2 cibles est installée. AWS CLI
3. Assurez-vous que chacune de vos instances Amazon EC2 cibles possède un profil d'instance IAM associé avec, au minimum, les autorisations `elasticloadbalancing : *` et `autoscaling : *`.
4. Incluez dans le répertoire de code source de votre application les scripts d'événement de cycle de vie de déploiement (`register_with_elb.sh`, `deregister_from_elb.sh` et `common_functions.sh`).
5. Dans la `appspec.yml` section pour la révision de l'application, fournissez des instructions CodeDeploy pour exécuter le `register_with_elb.sh` script pendant l'`ApplicationStart` événement et le `deregister_from_elb.sh` script pendant l'`ApplicationStop` événement.
6. Si l'instance fait partie d'un groupe Amazon EC2 Auto Scaling, vous pouvez ignorer cette étape.

Dans le script `common_functions.sh` :

- Si vous utilisez le [Classic Load Balancer](#), spécifiez les noms des équilibreurs de charge Elastic Load Balancing et apportez les modifications nécessaires aux autres paramètres de déploiement du fichier. `ELB_LIST=""`
- Si vous utilisez [Application Load Balancer ou Network Load Balancer](#), spécifiez les noms des groupes cibles Elastic Load Balancing `TARGET_GROUP_LIST=""` dans le fichier et apportez les modifications nécessaires aux autres paramètres de déploiement du fichier.

7. Groupez le code source de votre application, le `appspec.yml` et les scripts d'événement de cycle de vie de déploiement dans une révision d'application, puis chargez la révision. Déployez la révision sur les instances Amazon EC2. Pendant le déploiement, les scripts d'événements du cycle de vie du déploiement annuleront l'enregistrement de l'instance Amazon EC2 auprès de l'équilibreur de charge, attendront que la connexion soit épuisée, puis réenregistreront l'instance Amazon EC2 auprès de l'équilibreur de charge une fois le déploiement terminé.

Intégration avec les produits et services des partenaires

CodeDeploy intègre les produits et services partenaires suivants :

Ansible

Si vous possédez déjà un ensemble de playbooks [Ansible](#), mais que vous avez juste besoin d'un endroit où les exécuter, le modèle pour Ansible CodeDeploy montre comment quelques simples hooks de déploiement peuvent garantir la disponibilité d'Ansible sur l'instance de déploiement locale et l'exécution des playbooks. Si vous disposez déjà d'un processus de création et de gestion de votre inventaire, il existe également un module Ansible que vous pouvez utiliser pour installer et exécuter l'agent CodeDeploy.

En savoir plus :

- [Ansible et CodeDeploy](#)

Atlassian — Bamboo et Bitbucket

La tâche de CodeDeploy de [Bamboo](#) compresse le répertoire qui contient un AppSpec fichier dans un fichier .zip, télécharge le fichier sur Amazon S3, puis démarre le déploiement conformément à la configuration fournie dans l'application CodeDeploy.

La prise en charge d'Atlassian Bitbucket vous permet de transférer du code

vers des instances Amazon EC2 directement depuis l'interface utilisateur de Bitbucket, à la demande, vers l'un de vos groupes de déploiement. Cela signifie qu'après avoir mis à jour le code dans votre référentiel Bitbucket, vous n'avez pas besoin de vous connecter à votre plateforme d'intégration continue (CI) ou aux instances Amazon EC2 pour exécuter un processus de déploiement manuel.

En savoir plus :

- [Utilisation de la CodeDeploy tâche pour Bamboo](#)
- [Annonce du support d'Atlassian Bitbucket pour CodeDeploy](#)

Chef

AWS fournit deux exemples de modèles pour intégrer [Chef](#) et CodeDeploy. Le premier est un livre de recettes Chef qui installe et démarre l'agent CodeDeploy. Il vous permet de poursuivre la gestion de votre infrastructure hôte avec Chef tout en utilisant CodeDeploy. Le deuxième exemple de modèle montre comment CodeDeploy orchestrer l'exécution de livres de recettes et de recettes avec chef-solo sur chaque nœud.

En savoir plus :

- [Chef et CodeDeploy](#)

CircleCI	CircleCI fournit un ensemble d'outils de déploiement et d'intégration en continu et de test automatisé. Après avoir créé un rôle IAM AWS à utiliser avec CircleCI et configuré vos paramètres de déploiement dans votre fichier <code>circle.yml</code>, vous pouvez utiliser CodeDeploy CircleCI pour créer des révisions d'applications, les télécharger dans un compartiment Amazon S3, puis lancer et surveiller vos déploiements. En savoir plus : • Utilisation d'un CircleCI Orb pour déployer des applications sur AWS CodeDeploy
CloudBees	Vous pouvez utiliser le plugin CodeDeploy Jenkins, disponible sur CloudBeesDEV @cloud, comme action post-build. Par exemple, à la fin d'un pipeline de diffusion en continu, vous pouvez l'utiliser pour déployer une révision d'application sur votre flotte de serveurs. En savoir plus : • CodeDeploy Le plugin Jenkins est désormais disponible sur DEV @cloud
Codship	Vous pouvez utiliser Codship pour déployer des révisions d'applications via. CodeDeploy. Vous pouvez utiliser l'interface utilisateur Codship pour ajouter CodeDeploy à un pipeline de déploiement pour une succursale. En savoir plus : • Déployer vers CodeDeploy • CodeDeploy intégration sur Codship

GitHub

Vous pouvez utiliser CodeDeploy pour déployer des révisions d'applications à partir de [GitHub](#) référentiels. Vous pouvez également déclencher un déploiement à partir d'un GitHub référentiel chaque fois que le code source de ce référentiel est modifié.

En savoir plus :

- [Intégration CodeDeploy avec GitHub](#)
- [Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub](#)
- [Déployez automatiquement à GitHub partir de CodeDeploy](#)

HashiCorp Consul

Vous pouvez utiliser l'outil open source HashiCorp Consul pour garantir la santé et la stabilité de votre environnement d'applications lorsque vous y déployez des CodeDeploy applications. Vous pouvez utiliser Consul pour inscrire les applications à découvrir durant le déploiement, placer des applications et des nœuds en mode maintenance pour les omettre dans les déploiements et arrêter les déploiements si les instances cibles cessent d'être saines.

En savoir plus :

- [CodeDeploy déploiements avec Consul HashiCorp](#)

Jenkins

Le plugin CodeDeploy [Jenkins](#) fournit une étape de post-construction pour votre projet Jenkins. Une fois le build réussi, il comprime l'espace de travail, le télécharge sur Amazon S3 et lance un nouveau déploiement.

En savoir plus :

- [CodeDeployPlug-in Jenkins](#)
- [Configuration du plugin Jenkins pour CodeDeploy](#)

Puppet Labs

AWS fournit des exemples de modèles pour [Puppet et](#). CodeDeploy Le premier est un module Puppet qui installe et démarre l'agent CodeDeploy. Il vous permet de poursuivre la gestion de votre infrastructure hôte avec Puppet tout en utilisant CodeDeploy. Le deuxième exemple de modèle montre comment CodeDeploy orchestrer l'exécution de modules et de manifestes avec une marionnette masterless sur chaque nœud.

En savoir plus :

- [Marionnette et CodeDeploy](#)

SaltStack

Vous pouvez intégrer [SaltStack](#) l'infrastructure à CodeDeploy. Vous pouvez utiliser le CodeDeploy module pour installer et exécuter l'agent CodeDeploy sur vos serveurs ou, à l'aide de quelques crochets de déploiement simples, vous pouvez l'utiliser CodeDeploy pour orchestrer le fonctionnement de vos Salt States.

En savoir plus :

- [SaltStack et CodeDeploy](#)

TeamCity

Vous pouvez utiliser le plugin CodeDeploy Runner pour déployer des applications directement depuis TeamCity. Le plugin ajoute une TeamCity étape de génération qui prépare et télécharge une révision d'application dans un compartiment Amazon S3, enregistre la révision dans une CodeDeploy application, crée un CodeDeploy déploiement et, si vous le souhaitez, attend que le déploiement soit terminé.

En savoir plus :

- [CodeDeploy Runner \(Télécharger\)](#)
- [CodeDeploy Plug-in Runner \(documentation\)](#)

Travis CI

Vous pouvez configurer [Travis CI](#) pour déclencher un déploiement CodeDeploy après une compilation réussie.

En savoir plus :

- [Travis CI et déploiements CodeDeploy](#)

Rubriques

- [Intégration CodeDeploy avec GitHub](#)

Intégration CodeDeploy avec GitHub

CodeDeploy supports [GitHub](#), un service d'hébergement et de partage de code basé sur le Web. CodeDeploy peut déployer des révisions d'applications stockées dans GitHub des référentiels ou des compartiments Amazon S3 sur des instances. CodeDeploy prend en charge GitHub les déploiements EC2/sur site uniquement.

Rubriques

- [Déploiement de CodeDeploy révisions depuis GitHub](#)
- [GitHub comportements avec CodeDeploy](#)

Déploiement de CodeDeploy révisions depuis GitHub

Pour déployer une révision d'application depuis un GitHub référentiel vers des instances :

1. Créez une révision compatible avec le type CodeDeploy d'instance Amazon EC2 sur lequel vous allez effectuer le déploiement.

Pour créer une révision compatible, suivez les instructions fournies dans les rubriques [Planifier une révision pour CodeDeploy](#) et [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

2. Utilisez un GitHub compte pour ajouter votre révision à un GitHub dépôt.

Pour créer un GitHub compte, consultez la section [S'inscrire GitHub](#). Pour créer un GitHub dépôt, voir [Créer un dépôt](#).

3. Utilisez la page Créer un déploiement de la CodeDeploy console ou la AWS CLI create-deployment commande pour déployer votre révision depuis votre GitHub référentiel vers des instances cibles configurées pour être utilisées dans CodeDeploy des déploiements.

Si vous souhaitez appeler la create-deployment commande, vous devez d'abord utiliser la page Créer un déploiement de la console pour CodeDeploy autoriser l'interaction avec GitHub l'application spécifiée au nom de votre GitHub compte préféré. Cette opération ne doit être effectuée qu'une seule fois par application.

Pour savoir comment utiliser la page Créer un déploiement pour effectuer un déploiement à partir d'un GitHub référentiel, consultez [Créez un déploiement avec CodeDeploy](#).

Pour savoir comment appeler la create-deployment commande de déploiement à partir d'un GitHub référentiel, consultez [Création d'un déploiement de plate-forme de calcul EC2/sur site \(CLI\)](#).

Pour savoir comment préparer des instances en vue de leur utilisation dans les CodeDeploy déploiements, consultez [Utilisation d'instances pour CodeDeploy](#).

Pour de plus amples informations, veuillez consulter [Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub](#).

GitHub comportements avec CodeDeploy

Rubriques

- [GitHub authentification avec des applications dans CodeDeploy](#)
- [CodeDeploy interaction avec les GitHub référentiels privés et publics](#)
- [CodeDeploy interaction avec les référentiels gérés par l'organisation GitHub](#)
- [Déployez automatiquement à partir CodePipeline de CodeDeploy](#)

GitHub authentification avec des applications dans CodeDeploy

Une fois que vous avez donné CodeDeploy l'autorisation d'interagir avec GitHub, l'association entre ce GitHub compte et l'application est stockée dans CodeDeploy. Vous pouvez associer l'application à un autre GitHub compte. Vous pouvez également révoquer l'autorisation CodeDeploy d'interagir avec GitHub.

Pour associer un GitHub compte à une application dans CodeDeploy

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Choisissez l'application que vous souhaitez associer à un autre GitHub compte.
4. Si votre application ne possède pas de groupe de déploiement, choisissez Créer un groupe de déploiement pour en créer un. Pour de plus amples informations, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#). Un groupe de déploiement est nécessaire pour pouvoir choisir Créer un déploiement dans l'étape suivante.
5. Dans Déploiements, choisissez Créer un déploiement.

 Note

Vous n'avez pas à créer un nouveau déploiement. C'est actuellement le seul moyen de lier un autre GitHub compte à une application.

6. Dans Paramètres de déploiement, pour Type de révision, choisissez Mon application est stockée dans GitHub.
7. Effectuez l'une des actions suivantes :
 - Pour créer une connexion entre AWS CodeDeploy les applications et un GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Dans Nom du GitHub jeton, tapez un nom pour identifier cette connexion, puis choisissez Connect to GitHub. La page Web vous invite CodeDeploy à autoriser l'interaction avec GitHub votre application. Passez à l'étape 10.
 - Pour utiliser une connexion que vous avez déjà créée, dans Nom du GitHub jeton, sélectionnez son nom, puis choisissez Connect to GitHub. Passez à l'étape 8.
 - Pour créer une connexion à un autre GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Dans Nom du GitHub jeton, tapez un nom pour identifier la connexion, puis choisissez Connect to GitHub. La page Web vous invite CodeDeploy à autoriser l'interaction avec GitHub votre application. Passez à l'étape 10.
8. Si vous n'êtes pas encore connecté GitHub, suivez les instructions de la page de connexion pour vous connecter avec le GitHub compte auquel vous souhaitez associer l'application.

9. Choisissez Autoriser l'application. GitHub donne CodeDeploy l'autorisation d'interagir avec GitHub l'application sélectionnée au nom du GitHub compte connecté.
10. Si vous ne souhaitez pas créer de déploiement, choisissez Annuler.

Pour révoquer l'autorisation CodeDeploy d'interagir avec GitHub

1. Connectez-vous à [GitHub](#) l'aide des informations d'identification du GitHub compte pour lequel vous souhaitez révoquer AWS CodeDeploy l'autorisation.
2. Ouvrez la page GitHub [Applications](#), CodeDeploy localisez-la dans la liste des applications autorisées, puis suivez la GitHub procédure de révocation de l'autorisation d'une application.

CodeDeploy interaction avec les GitHub référentiels privés et publics

CodeDeploy prend en charge le déploiement d'applications à partir de GitHub référentiels privés et publics. Lorsque vous CodeDeploy autorisez l'accès en votre GitHub nom, vous CodeDeploy aurez un accès en lecture-écriture à tous les GitHub référentiels privés auxquels votre GitHub compte a accès. Cependant, CodeDeploy seules les lectures se trouvent dans GitHub les référentiels. Il n'écrit dans aucun de vos GitHub dépôts privés.

CodeDeploy interaction avec les référentiels gérés par l'organisation GitHub

Par défaut, GitHub les référentiels gérés par une organisation (par opposition aux référentiels privés ou publics de votre compte) n'accordent pas l'accès aux applications tierces, notamment. CodeDeploy Votre déploiement échouera si les restrictions relatives aux applications tierces d'une organisation sont activées GitHub et que vous tentez de déployer du code depuis son GitHub référentiel. Il existe deux façons de résoudre ce problème.

- En tant que membre de l'organisation, vous pouvez demander au propriétaire de l'organisation d'approuver l'accès à CodeDeploy. Les étapes à suivre pour demander cet accès varient selon que vous avez déjà autorisé CodeDeploy ou non votre compte individuel :
 - Si vous avez un accès autorisé CodeDeploy à votre compte, consultez la section [Demander l'approbation de l'organisation pour vos applications autorisées](#).
 - Si vous n'êtes pas encore autorisé à accéder CodeDeploy à votre compte, consultez la section [Demande d'approbation de l'organisation pour les applications tierces](#).
- Le propriétaire de l'organisation peut désactiver toutes les restrictions d'application tierce pour l'organisation. Pour plus d'informations, consultez la section [Désactivation des restrictions relatives aux applications tierces pour votre organisation](#).

Pour plus d'informations, voir [À propos des restrictions relatives aux applications tierces](#).

Déployez automatiquement à partir CodePipeline de CodeDeploy

Vous pouvez déclencher un déploiement à partir de CodePipeline chaque fois que le code source change. Pour plus d'informations, consultez [CodePipeline](#).

Exemples d'intégration issus de la communauté

Les sections suivantes fournissent des liens vers des billets de blogs, des articles et des exemples fournis par la communauté.

Note

Ces liens sont fournis uniquement à titre informatif et ne doivent pas être considérés comme une liste complète ni comme une approbation du contenu des exemples. AWS n'est pas responsable du contenu ni de l'exactitude du contenu externe.

Billets de blogs

- [Automatisation du CodeDeploy provisionnement dans CloudFormation](#)

Découvrez comment configurer le déploiement d'une application en CodeDeploy utilisant CloudFormation.

Publié en janvier 2016

- [AWS Toolkit for Eclipse Intégration avec CodeDeploy \(partie 1\)](#)

[AWS Toolkit for Eclipse Intégration avec CodeDeploy \(partie 2\)](#)

[AWS Toolkit for Eclipse Intégration avec CodeDeploy \(partie 3\)](#)

Découvrez comment les développeurs Java peuvent utiliser le CodeDeploy plugin pour Eclipse pour déployer des applications Web AWS directement depuis les environnements de développement Eclipse.

Publié en février 2015

- [Déployez automatiquement à GitHub partir de CodeDeploy](#)

Découvrez comment les déploiements automatiques de GitHub à CodeDeploy peuvent être utilisés pour créer un end-to-end pipeline, du contrôle de source à vos environnements de test ou de production.

Publié en décembre 2014

CodeDeploy tutoriels

Cette section inclut des didacticiels qui vous aideront à apprendre à utiliser CodeDeploy.

Les procédures décrites dans ces didacticiels fournissent des suggestions concernant l'emplacement dans lequel stocker les fichiers (par exemple, c:\temp) et les noms à attribuer aux compartiments, sous-dossiers ou fichiers (par exemple, amzn-s3-demo-bucket et CodeDeployDemo - EC2 -trust.json, respectivement) HelloWorldApp, mais vous n'êtes pas obligé de les utiliser. Veuillez simplement à substituer vos noms et emplacements de fichiers lorsque vous exécutez les procédures.

Rubriques

- [Tutoriel : Déploiement WordPress sur une instance Amazon EC2 \(Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix\)](#)
- [Tutoriel : Déployez un message « Bonjour tout le monde ! » application avec CodeDeploy \(Windows Server\)](#)
- [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#)
- [Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling](#)
- [Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub](#)
- [Tutoriel : Déployer une application dans Amazon ECS](#)
- [Tutoriel : Déployer un service Amazon ECS avec un test de validation](#)
- [Tutoriel : Déployer une fonction Lambda mise à jour avec CodeDeploy et le modèle d'application AWS sans serveur](#)

Tutoriel : Déploiement WordPress sur une instance Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix)

Dans ce didacticiel, vous allez déployer WordPress un outil de blog open source et un système de gestion de contenu basés sur PHP et MySQL sur une seule instance Amazon EC2 exécutant Amazon Linux ou Red Hat Enterprise Linux (RHEL).

Ce n'est pas ce que vous recherchez ?

- Pour vous entraîner à déployer sur une instance Amazon EC2 exécutant plutôt Windows Server, consultez. [Tutoriel : Déployez un message « Bonjour tout le monde ! » application avec CodeDeploy \(Windows Server\)](#)
- Pour vous entraîner à déployer sur une instance sur site plutôt que sur une instance Amazon EC2, consultez. [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#)

Les étapes de ce didacticiel sont présentées du point de vue d'une machine de développement locale exécutant Linux, macOS ou Unix. Bien que vous puissiez effectuer la plupart de ces étapes sur une machine locale exécutant Windows, vous devez adapter les étapes qui couvrent des commandes telles que `chmod` et `wget`, des applications telles que `sed` et des chemins de répertoires tels que `/tmp`.

Avant de commencer ce didacticiel, vous devez remplir les conditions préalables dans [Commencer avec CodeDeploy](#). Il s'agit notamment de configurer un utilisateur, d'installer ou de mettre à niveau le AWS CLI, ainsi que de créer un profil d'instance IAM et un rôle de service.

Rubriques

- [Étape 1 : Lancer et configurer une instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux](#)
- [Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux](#)
- [Étape 3 : Chargez votre WordPress application sur Amazon S3](#)
- [Étape 4 : Déployez votre WordPress application](#)
- [Étape 5 : Mettez à jour et redéployez votre application WordPress](#)
- [Étape 6 : Nettoyez votre WordPress application et les ressources associées](#)

Étape 1 : Lancer et configurer une instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux

Pour déployer l' WordPress application CodeDeploy, vous aurez besoin d'une instance Amazon EC2 exécutant Amazon Linux ou Red Hat Enterprise Linux (RHEL). L'instance Amazon EC2 nécessite une nouvelle règle de sécurité entrante qui autorise les connexions HTTP. Cette règle est nécessaire pour afficher la WordPress page dans un navigateur après son déploiement réussi.

Suivez les instructions de la section [Créez une instance Amazon EC2 pour CodeDeploy](#). Lorsque vous arriverez à la partie de ces instructions concernant l'attribution d'une balise d'instance Amazon EC2 à l'instance, assurez-vous de spécifier la clé **Name** de balise et la valeur de balise de **CodeDeployDemo** (Si vous spécifiez une clé de balise ou une valeur de balise différente, les instructions fournies dans [Étape 4 : Déployez votre WordPress application](#) peuvent générer des résultats inattendus.)

Après avoir suivi les instructions pour lancer l'instance Amazon EC2, revenez à cette page et passez à la section suivante. Ne passez pas à [Créez une application avec CodeDeploy](#) l'étape suivante.

Connectez-vous à votre instance Amazon Linux ou RHEL Amazon EC2

Après le lancement de votre nouvelle instance Amazon EC2, suivez ces instructions pour vous entraîner à vous y connecter.

1. Utilisez la ssh commande (ou un émulateur de terminal compatible SSH tel que [PuTTY](#)) pour vous connecter à votre instance Amazon Linux ou RHEL Amazon EC2. Vous aurez besoin de l'adresse DNS publique de l'instance et de la clé privée de la paire de clés que vous avez utilisée lorsque vous avez démarré l'instance Amazon EC2. Pour plus d'informations, consultez [Connect to Your Instance](#).

Par exemple, si l'adresse DNS publique est

ec2-01-234-567-890.compute-1.amazonaws.com et que votre paire de clés d'instance Amazon EC2 pour l'accès SSH est nommée **codedeploydemo.pem**, vous devez taper :

```
ssh -i /path/to/codedeploydemo.pem ec2-  
user@ec2-01-234-567-890.compute-1.amazonaws.com
```

Remplacez **/path/to/codedeploydemo.pem** par le chemin de votre **.pem** fichier et l'exemple d'adresse DNS par l'adresse de votre instance Amazon Linux ou RHEL Amazon EC2.

Note

Si vous recevez une erreur indiquant que les autorisations de votre fichier de clé sont trop ouvertes, vous devez limiter ces autorisations pour accorder l'accès uniquement à l'utilisateur actuel (vous-même). Par exemple, avec la **chmod** commande sous Linux, macOS ou Unix, tapez :

```
chmod 400 /path/to/codedeploydemo.pem
```

2. Une fois connecté, vous verrez la bannière AMI de l'instance Amazon EC2. Pour Amazon Linux, cela devrait ressembler à ceci :

```

 _|  _|_ )
 _| (    /  Amazon Linux AMI
 _|\_|_|_|

```

3. Vous pouvez désormais vous déconnecter de l'instance Amazon EC2 en cours d'exécution.

Warning

N'arrêtez pas ou ne mettez pas fin à l'instance Amazon EC2. Sinon, je CodeDeploy ne pourrai pas y déployer.

Ajoutez une règle entrante qui autorise le trafic HTTP vers votre instance Amazon Linux ou RHEL Amazon EC2

L'étape suivante confirme que votre instance Amazon EC2 possède un port HTTP ouvert afin que vous puissiez voir la page d'accueil de l' WordPress application déployée dans un navigateur.

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
2. Choisissez Instances, puis choisissez votre instance.
3. Dans l'onglet Description, sous Groupes de sécurité, choisissez Afficher les règles entrantes.

Vous devriez voir une liste de règles dans votre groupe de sécurité, comme suit :

```

Security Groups associated with i-1234567890abcdef0
Ports      Protocol    Source      launch-wizard-N
22         tcp         0.0.0.0/0   #

```

4. Sous Groupes de sécurité, choisissez le groupe de sécurité pour votre instance Amazon EC2. Il pourrait être nommé **launch-wizard-*N***. Le ***N*** dans le nom est un numéro attribué à votre groupe de sécurité lorsque l'instance a été créée.

Choisissez l'onglet Entrant. Si le groupe de sécurité de votre instance est correctement configuré, vous devriez voir apparaître une règle avec les valeurs suivantes :

- Type : HTTP
 - Protocole : TCP
 - Plage de ports : 80
 - Source : 0.0.0,0/0
5. Si aucune règle ne contient ces valeurs, utilisez les procédures décrites dans [Ajouter des règles à un groupe de sécurité](#) pour les ajouter à une nouvelle règle de sécurité.

Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux

Désormais, il est temps de configurer le contenu source de votre application pour que vous ayez quelque chose à déployer sur l'instance.

Rubriques

- [Obtenez le code source](#)
- [Créez des scripts pour exécuter votre application](#)
- [Ajouter un fichier de spécification d'application](#)

Obtenez le code source

Dans le cadre de ce didacticiel, vous déployez la plateforme de publication de WordPress contenu depuis votre machine de développement vers l'instance Amazon EC2 cible. Pour obtenir le code WordPress source, vous pouvez utiliser les appels de ligne de commande intégrés. Ou, si Git est installé sur votre machine de développement, vous pouvez l'utiliser à la place.

Pour ces étapes, nous supposons que vous avez téléchargé une copie du code WordPress source /tmp dans le répertoire de votre machine de développement. (Vous pouvez choisir n'importe quel répertoire, mais n'oubliez pas de remplacer /tmp par votre emplacement chaque fois que ce répertoire est spécifié dans ces étapes.)

Choisissez l'une des deux options suivantes pour copier les fichiers WordPress source sur votre machine de développement. La première option utilise des appels de ligne de commande intégrés. La deuxième option utilise Git.

Rubriques

- [Pour obtenir une copie du code WordPress source \(appels de ligne de commande intégrés\)](#)
- [Pour obtenir une copie du code WordPress source \(Git\)](#)

Pour obtenir une copie du code WordPress source (appels de ligne de commande intégrés)

1. Appelez la `wget` commande pour télécharger une copie du code WordPress source, sous forme de fichier `.zip`, dans le répertoire actuel :

```
wget https://github.com/WordPress/WordPress/archive/master.zip
```

2. Appelez les commandes `unzip`, `mkdir`, `cp` et `rm` pour :
 - Décompressez le fichier `master.zip` dans le répertoire (dossier) `/tmp/WordPress_Temp`.
 - Copiez son contenu décompressé dans le dossier de destination `/tmp/WordPress`.
 - Supprimez le dossier `/tmp/WordPress_Temp` temporaire et le fichier `master`.

Exécutez une par une les commandes :

```
unzip master -d /tmp/WordPress_Temp
```

```
mkdir -p /tmp/WordPress
```

```
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
```

```
rm -rf /tmp/WordPress_Temp
```

```
rm -f master
```

Cela vous laisse un ensemble propre de fichiers de code WordPress source dans le `/tmp/WordPress` dossier.

Pour obtenir une copie du code WordPress source (Git)

1. Téléchargez et installez [Git](#) sur votre machine de développement.
2. Dans le dossier `/tmp/WordPress`, appelez la commande `git init`.
3. Appelez la `git clone` commande pour cloner le WordPress dépôt public, en créant votre propre copie dans le dossier de `/tmp/WordPress` destination :

```
git clone https://github.com/WordPress/WordPress.git /tmp/WordPress
```

Cela vous laisse un ensemble propre de fichiers de code WordPress source dans le `/tmp/WordPress` dossier.

Créez des scripts pour exécuter votre application

Créez ensuite un dossier et des scripts dans le répertoire. CodeDeploy utilise ces scripts pour configurer et déployer la révision de votre application sur l'instance Amazon EC2 cible. Vous pouvez utiliser un éditeur de texte quelconque pour créer ces scripts.

1. Créez un répertoire de scripts dans votre copie du code WordPress source :

```
mkdir -p /tmp/WordPress/scripts
```

2. Créez un fichier `install_dependencies.sh` dans `/tmp/WordPress/scripts`. Ajoutez les lignes suivantes dans le fichier. Ce script `install_dependencies.sh` installe Apache, MySQL et PHP. Il ajoute également la prise en charge de MySQL à PHP.

```
#!/bin/bash
sudo amazon-linux-extras install php7.4
sudo yum install -y httpd mariadb-server php
```

3. Créez un fichier `start_server.sh` dans `/tmp/WordPress/scripts`. Ajoutez les lignes suivantes dans le fichier. Ce script `start_server.sh` démarre Apache et MySQL.

```
#!/bin/bash
systemctl start mariadb.service
systemctl start httpd.service
systemctl start php-fpm.service
```

4. Créez un fichier `stop_server.sh` dans `/tmp/WordPress/scripts`. Ajoutez les lignes suivantes dans le fichier. Ce script `stop_server.sh` arrête Apache et MySQL.

```
#!/bin/bash
isExistApp="pgrep httpd"
if [[ -n $isExistApp ]]; then
systemctl stop httpd.service
fi
isExistApp=pgrep mysqld
if [[ -n $isExistApp ]]; then
systemctl stop mariadb.service
fi
isExistApp=pgrep php-fpm
if [[ -n $isExistApp ]]; then
systemctl stop php-fpm.service
fi
```

5. Créez un fichier `create_test_db.sh` dans `/tmp/WordPress/scripts`. Ajoutez les lignes suivantes dans le fichier. Ce `create_test_db.sh` script utilise MySQL pour créer une **test** base de données WordPress à utiliser.

```
#!/bin/bash
mysql -uroot <<CREATE_TEST_DB
CREATE DATABASE IF NOT EXISTS test;
CREATE_TEST_DB
```

6. Enfin, créez un script `change_permissions.sh` dans `/tmp/WordPress/scripts`. Il permet de modifier les autorisations des dossiers dans Apache.

Important

Ce script a mis à jour les autorisations sur le dossier `/tmp/WordPress` pour que tout le monde puisse écrire dans celui-ci. Cela est nécessaire pour WordPress pouvoir écrire dans sa base de données pendant [Étape 5 : Mettez à jour et redéployez votre application WordPress](#). Une fois l'WordPress application configurée, exécutez la commande suivante pour mettre à jour les autorisations selon un paramètre plus sécurisé :

```
chmod -R 755 /var/www/html/WordPress
```

```
#!/bin/bash
chmod -R 777 /var/www/html/WordPress
```

7. Fournissez à tous les scripts des autorisations d'exécution. Sur la ligne de commande, tapez :

```
chmod +x /tmp/WordPress/scripts/*
```

Ajouter un fichier de spécification d'application

Ajoutez ensuite un fichier de spécification d'application (AppSpec fichier), un fichier au format [YAML](#) utilisé pour : CodeDeploy

- Mappez les fichiers source de la révision de votre application à leurs destinations sur l'instance Amazon EC2 cible.
- spécifier des autorisations personnalisées pour les fichiers déployés ;
- Spécifiez les scripts à exécuter sur l'instance Amazon EC2 cible lors du déploiement.

Le AppSpec fichier doit être nommé `appspec.yml`. Il doit être placé dans le répertoire racine du code source de l'application. Dans ce didacticiel, le répertoire racine est `/tmp/WordPress`.

Grâce à votre éditeur de texte, créez un fichier nommé `appspec.yml`. Ajoutez les lignes suivantes dans le fichier :

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
```



```
- location: scripts/change_permissions.sh
  timeout: 300
  runas: root
ApplicationStart:
- location: scripts/start_server.sh
- location: scripts/create_test_db.sh
  timeout: 300
  runas: root
ApplicationStop:
- location: scripts/stop_server.sh
  timeout: 300
  runas: root
```

CodeDeploy utilise ce AppSpec fichier pour copier tous les fichiers du /tmp/WordPress dossier de la machine de développement vers le /var/www/html/WordPress dossier de l'instance Amazon EC2 cible. Au cours du déploiement, CodeDeploy exécute les scripts spécifiés tels qu'ils root figurent dans le /var/www/html/WordPress/scripts dossier de l'instance Amazon EC2 cible lors d'événements spécifiques du cycle de vie du déploiement, tels que **BeforeInstall** et **AfterInstall**. Si l'exécution de l'un de ces scripts prend plus de 300 secondes (5 minutes), CodeDeploy arrête le déploiement et marque le déploiement comme ayant échoué.

Pour plus d'informations sur ces paramètres, consultez la rubrique [CodeDeploy AppSpec référence de fichier](#).

Important

Les emplacements et les nombres d'espaces entre les différents éléments de ce fichier sont importants. Si l'espacement est incorrect, CodeDeploy cela génère une erreur qui peut être difficile à déboguer. Pour de plus amples informations, veuillez consulter [AppSpec Espacement des fichiers](#).

Étape 3 : Chargez votre WordPress application sur Amazon S3

Vous allez maintenant préparer et télécharger votre contenu source vers un emplacement à partir duquel CodeDeploy vous pourrez le déployer. Les instructions suivantes vous montrent comment configurer un compartiment Amazon S3, préparer les fichiers de révision de l'application pour le compartiment, regrouper les fichiers de révision, puis transférer la révision dans le compartiment.

 Note

Bien que cela ne soit pas abordé dans ce didacticiel, vous pouvez l'utiliser CodeDeploy pour déployer des applications depuis des GitHub référentiels vers des instances. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy avec GitHub](#).

Rubriques

- [Provisionner un compartiment Amazon S3](#)
- [Préparer les fichiers de l'application pour le bucket](#)
- [Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive](#)

Provisionner un compartiment Amazon S3

Créez un conteneur ou un compartiment de stockage dans Amazon S3 ou utilisez un compartiment existant. Assurez-vous que vous pouvez télécharger la révision dans le compartiment et que les instances Amazon EC2 utilisées dans les déploiements peuvent télécharger la révision depuis le compartiment.

Vous pouvez utiliser AWS CLI la console Amazon S3 ou Amazon S3 APIs pour créer un compartiment Amazon S3. Après avoir créé le bucket, assurez-vous de donner des autorisations d'accès au bucket et à votre AWS compte.

 Note

Les noms de compartiment doivent être uniques sur Amazon S3 pour tous les AWS comptes. Si vous n'êtes pas en mesure d'utiliser **amzn-s3-demo-bucket**, essayez un autre nom de compartiment, tel que **amzn-s3-demo-bucket**, suivi d'un tiret et de vos initiales ou d'un autre identificateur unique. Veuillez ensuite à substituer votre nom de compartiment à **amzn-s3-demo-bucket** chaque fois que vous le verrez tout au long de ce didacticiel.

Le compartiment Amazon S3 doit être créé dans la même AWS région que celle où vos instances Amazon EC2 cibles sont lancées. Par exemple, si vous créez le bucket dans la région USA Est (Virginie du Nord), vos instances Amazon EC2 cibles doivent être lancées dans la région USA Est (Virginie du Nord).

Rubriques

- [Pour créer un compartiment Amazon S3 \(CLI\)](#)
- [Pour créer un compartiment Amazon S3 \(console\)](#)
- [Donnez des autorisations au compartiment et au AWS compte Amazon S3](#)

Pour créer un compartiment Amazon S3 (CLI)

Appelez la mb commande pour créer un compartiment Amazon S3 nommé **amzn-s3-demo-bucket** :

```
aws s3 mb s3://amzn-s3-demo-bucket --region region
```

Pour créer un compartiment Amazon S3 (console)

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Dans la console Amazon S3, choisissez Create bucket.
3. Dans la zone Nom du compartiment, saisissez un nom pour le compartiment.
4. Dans la liste Région, choisissez la région cible, puis Créer.

Donnez des autorisations au compartiment et au AWS compte Amazon S3

Vous devez disposer des autorisations nécessaires pour effectuer le téléchargement dans le compartiment Amazon S3. Vous pouvez spécifier ces autorisations par le biais d'une politique de compartiment Amazon S3. Par exemple, dans la politique de compartiment Amazon S3 suivante, l'utilisation du caractère générique (*) permet 111122223333 au AWS compte de télécharger des fichiers dans n'importe quel répertoire du compartiment Amazon S3 nommé amzn-s3-demo-bucket :

```
{
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Principal": {
        "AWS": [
```

```

        "111122223333"
      ]
    }
  ]
}

```

Pour consulter l'identifiant de votre AWS compte, consultez la section [Trouver votre identifiant de AWS compte](#).

C'est le bon moment pour vérifier que le compartiment Amazon S3 autorisera les demandes de téléchargement provenant de chaque instance Amazon EC2 participante. Vous pouvez le spécifier par le biais d'une politique de compartiment Amazon S3. Par exemple, dans la politique de compartiment Amazon S3 suivante, l'utilisation du caractère générique (*) permet à toute instance Amazon EC2 associée à un profil d'instance IAM contenant l'arn:aws:iam::444455556666:role/CodeDeployDemoARN de télécharger des fichiers depuis n'importe quel répertoire du compartiment Amazon S3 nommé : amzn-s3-demo-bucket

```

{
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::444455556666:role/CodeDeployDemo"
        ]
      }
    }
  ]
}

```

Pour plus d'informations sur la façon de générer et d'associer une politique de compartiment Amazon S3, consultez les [exemples de politique de compartiment](#).

Pour plus d'informations sur la création et l'attachement d'une stratégie IAM, consultez la section [Utilisation des politiques](#).

Préparer les fichiers de l'application pour le bucket

Assurez-vous que les fichiers WordPress d'application, le AppSpec fichier et les scripts sont organisés sur votre machine de développement de la manière suivante :

```
/tmp/  
|--WordPress/  
    |-- appspec.yml  
    |-- scripts/  
        |-- change_permissions.sh  
        |-- create_test_db.sh  
        |-- install_dependencies.sh  
        |-- start_server.sh  
        |-- stop_server.sh  
    |-- wp-admin/  
        |-- (various files...)  
    |-- wp-content/  
        |-- (various files...)  
    |-- wp-includes/  
        |-- (various files...)  
    |-- index.php  
    |-- license.txt  
    |-- readme.html  
    |-- (various files ending with .php...)
```

Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive

Regroupez les fichiers WordPress d'application et le AppSpec fichier dans un fichier d'archive (connu sous le nom de révision de l'application).

Note

Il est possible qu'il vous soit facturé le stockage des objets dans un compartiment et le transfert des révisions d'application vers et hors d'un compartiment. Pour plus d'informations, consultez [Tarification Amazon S3](#).

1. Sur la machine de développement, basculez vers le dossier dans lequel les fichiers sont stockés :

```
cd /tmp/WordPress
```

Note

Si vous ne basculez pas dans ce dossier, la création d'un bundle de fichiers commencera dans votre dossier actif. Par exemple, si votre dossier actif est /tmp au lieu de /tmp/WordPress, la création du bundle commencera avec les fichiers et les sous-dossiers figurant dans le dossier tmp, lequel peut inclure plus que le simple sous-dossier WordPress.

2. Appelez la commande `create-application` pour enregistrer une nouvelle application nommée **WordPress_App** :

```
aws deploy create-application --application-name WordPress_App
```

3. Appelez la commande CodeDeploy [push](#) pour regrouper les fichiers, télécharger les révisions CodeDeploy sur Amazon S3 et enregistrer les informations relatives à la révision téléchargée, le tout en une seule action.

```
aws deploy push \  
  --application-name WordPress_App \  
  --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \  
  --ignore-hidden-files
```

Cette commande regroupe les fichiers du répertoire actuel (à l'exception des fichiers cachés) dans un seul fichier d'archive nommé **WordPressApp.zip**, télécharge la révision dans le **amzn-s3-demo-bucket** compartiment et enregistre les informations relatives à CodeDeploy la révision téléchargée.

Étape 4 : Déployez votre WordPress application

Vous devez maintenant déployer l'exemple de révision WordPress d'application que vous avez chargé sur Amazon S3. Vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour déployer la révision et suivre la progression du déploiement. Une fois que la révision d'application est déployée avec succès, vous vérifiez les résultats.

Rubriques

- [Déployez la révision de votre application avec CodeDeploy](#)
- [Surveillez et résolvez les problèmes liés à votre déploiement](#)
- [Vérifiez votre déploiement](#)

Déployez la révision de votre application avec CodeDeploy

Utilisez la console AWS CLI ou la console pour déployer la révision de votre application.

Rubriques

- [Pour déployer votre révision d'application \(interface CLI\)](#)
- [Pour déployer votre révision d'application \(console\)](#)

Pour déployer votre révision d'application (interface CLI)

1. Le déploiement a besoin d'un groupe de déploiement. Toutefois, avant de créer le groupe de déploiement, vous avez besoin d'un nom ARN de rôle de service. Un rôle de service est un rôle IAM qui autorise un service à agir en votre nom. Dans ce cas, le rôle de service donne CodeDeploy l'autorisation d'accéder à vos instances Amazon EC2 pour étendre (lire) leurs balises d'instance Amazon EC2.

Vous devriez avoir déjà suivi les instructions de la rubrique [Création d'un rôle de service \(CLI\)](#) pour créer un rôle de service. Pour obtenir le nom ARN du rôle de service, consultez la page [Obtenez l'ARN du rôle de service \(CLI\)](#).

2. Maintenant que vous disposez de l'ARN du rôle de service, appelez la `create-deployment-group` commande pour créer un groupe de déploiement nommé **WordPress_DepGroup**, associé à l'application nommée **WordPress_App**, à l'aide de la balise Amazon EC2 **CodeDeployDemo** et de la configuration de déploiement nommée : **CodeDeployDefault.OneAtATime**

```
aws deploy create-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DepGroup \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \  
  --service-role-arn serviceRoleARN
```

 Note

La [create-deployment-group](#) commande permet de créer des déclencheurs qui entraînent l'envoi de notifications Amazon SNS aux abonnés du topic concernant des événements spécifiques liés aux déploiements et aux instances. La commande prend également en charge des options permettant d'annuler automatiquement les déploiements et de configurer des alarmes pour arrêter les déploiements lorsque les seuils de surveillance des CloudWatch alarmes d'Amazon sont atteints. Les commandes associées à ces actions ne sont pas incluses dans ce didacticiel.

3. Avant de créer un déploiement, l' CodeDeploy agent doit être installé sur les instances de votre groupe de déploiement. Vous pouvez installer l'agent à partir de la ligne de commande AWS Systems Manager avec la commande suivante :

```
aws ssm create-association \  
  --name AWS-ConfigureAWSPackage \  
  --targets Key=tag:Name,Values=CodeDeployDemo \  
  --parameters action=Install,name=AWSCodeDeployAgent \  
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Cette commande crée une association dans Systems Manager State Manager qui installera l' CodeDeploy agent puis tentera de le mettre à jour à 2h00 tous les dimanches matins. Pour plus d'informations sur l' CodeDeploy agent, consultez la section [Utilisation de l' CodeDeploy agent](#). Pour plus d'informations sur Systems Manager, consultez [What is AWS Systems Manager](#).

4. Maintenant, appelez la commande create-deployment pour créer un déploiement associé à l'application nommée **WordPress_App**, à la configuration de déploiement nommée **CodeDeployDefault.OneAtATime** et au groupe de déploiement nommé **WordPress_DepGroup**, en utilisant la révision d'application nommée **WordPressApp.zip** dans le compartiment nommé **amzn-s3-demo-bucket** :

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DepGroup \  
  --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=WordPressApp.zip
```


Pour déployer votre révision d'application (console)

1. Avant d'utiliser la CodeDeploy console pour déployer la révision de votre application, vous avez besoin d'un ARN de rôle de service. Un rôle de service est un rôle IAM qui autorise un service à agir en votre nom. Dans ce cas, le rôle de service donne CodeDeploy l'autorisation d'accéder à vos instances Amazon EC2 pour étendre (lire) leurs balises d'instance Amazon EC2.

Vous devriez avoir déjà suivi les instructions de la rubrique [Création d'un rôle de service \(console\)](#) pour créer un rôle de service. Pour obtenir le nom ARN du rôle de service, consultez la page [Obtenir l'ARN du rôle de service \(console\)](#).

2. Maintenant que vous avez l'ARN, utilisez la CodeDeploy console pour déployer la révision de votre application :

Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

3. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
4. Dans la liste des applications, choisissez WordPress_App.
5. Dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement.
6. Dans Nom du groupe de déploiement, saisissez **WordPress_DepGroup**.
7. Sous Deployment type, choisissez déploiement In-place.
8. Dans Configuration de l'environnement, sélectionnez les instances Amazon EC2.
9. Dans la configuration de l'agent avec AWS Systems Manager, conservez les valeurs par défaut.
10. Dans Clé, entrez **Name**.
11. Dans Value (Valeur), entrez **CodeDeployDemo**.

Note

Une fois que vous avez saisi le code **CodeDeployDemo**, un 1 doit apparaître sous Instances correspondantes pour confirmer qu'une instance Amazon EC2 correspondante CodeDeploy a été trouvée.

12. Dans Configuration du déploiement, choisissez CodeDeployDefault. OneAtATime.
13. Dans ARN de rôle de service, choisissez l'ARN du rôle de service, puis choisissez Créer un groupe de déploiement.
14. Choisissez Créer un déploiement.
15. Dans Groupe de déploiement, choisissez **WordPress_DepGroup**.
16. À côté de Type de référentiel, choisissez Mon application est stockée dans Amazon S3. Dans Emplacement de la révision, entrez l'emplacement de l'exemple de révision de WordPress l'application que vous avez précédemment chargé sur Amazon S3. Pour obtenir cet emplacement :
 - a. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
 - b. Dans la liste des buckets, choisissez amzn-s3-demo-bucket (ou le nom du bucket dans lequel vous avez chargé la révision de votre application).
 - c. Dans la liste des objets, choisissez WordPressApp.zip.
 - d. Dans l'onglet Présentation, copiez la valeur du champ Lien dans votre presse-papiers.

Elle peut ressembler à ceci :

`https://s3.amazonaws.com/amzn-s3-demo-bucket/WordPressApp.zip`
 - e. Retournez à la CodeDeploy console, puis dans Emplacement de la révision, collez la valeur du champ Lien.
17. Dans la liste File type (Type de fichier), si un message s'affiche indiquant que le type de fichier n'a pas pu être détecté, choisissez .zip.
18. (Facultatif) Tapez un commentaire dans la zone Deployment description.
19. Développez les remplacements des groupes de déploiement, puis sélectionnez CodeDeployDefault Configuration du déploiement. OneAtATime.
20. Choisissez Démarrer le déploiement. Des informations sur votre déploiement nouvellement créé s'affichent dans la page Deployments.

Surveillez et résolvez les problèmes liés à votre déploiement

Utilisez la console AWS CLI ou la console pour surveiller et résoudre les problèmes liés à votre déploiement.

Rubriques

- [Pour surveiller et déboguer votre déploiement \(interface CLI\)](#)
- [Pour surveiller et déboguer votre déploiement \(console\)](#)

Pour surveiller et déboguer votre déploiement (interface CLI)

1. Obtenez l'ID du déploiement en appelant la commande `list-deployments` sur l'application nommée **WordPress_App** et le groupe de déploiement nommé **WordPress_DepGroup** :

```
aws deploy list-deployments --application-name WordPress_App --deployment-group-name WordPress_DepGroup --query 'deployments' --output text
```

2. Appelez la commande `get-deployment` avec l'ID du déploiement :

```
aws deploy get-deployment --deployment-id deploymentID --query 'deploymentInfo.status' --output text
```

3. La commande renvoie l'état général du déploiement. En cas de réussite, la valeur est `Succeeded`.

Si l'état général est `Failed`, vous pouvez appeler des commandes telles que [list-deployment-instances](#) et [get-deployment-instance](#) pour résoudre les problèmes. Pour plus d'options de débogage, consultez la page [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#).

Pour surveiller et déboguer votre déploiement (console)

Sur la page Déploiements de la CodeDeploy console, vous pouvez surveiller l'état de votre déploiement dans la colonne État.

Pour obtenir plus d'informations sur votre déploiement, notamment si la valeur de la colonne État a une valeur différente de `Succeeded` :

1. Dans la table Déploiements, choisissez du nom du déploiement. Une fois qu'un déploiement a échoué, un message décrivant la raison de l'échec s'affiche.
2. Dans Activité des instances, d'autres informations sur le déploiement s'affichent. Après l'échec d'un déploiement, vous pourrez peut-être déterminer sur quelles instances Amazon EC2 et à quelle étape le déploiement a échoué.

3. Si vous souhaitez effectuer davantage d'opérations de dépannage, vous pouvez utiliser une technique telle que celle décrite dans [View Instance Details](#). Vous pouvez également analyser les fichiers journaux de déploiement sur une instance Amazon EC2. Pour de plus amples informations, veuillez consulter [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#).

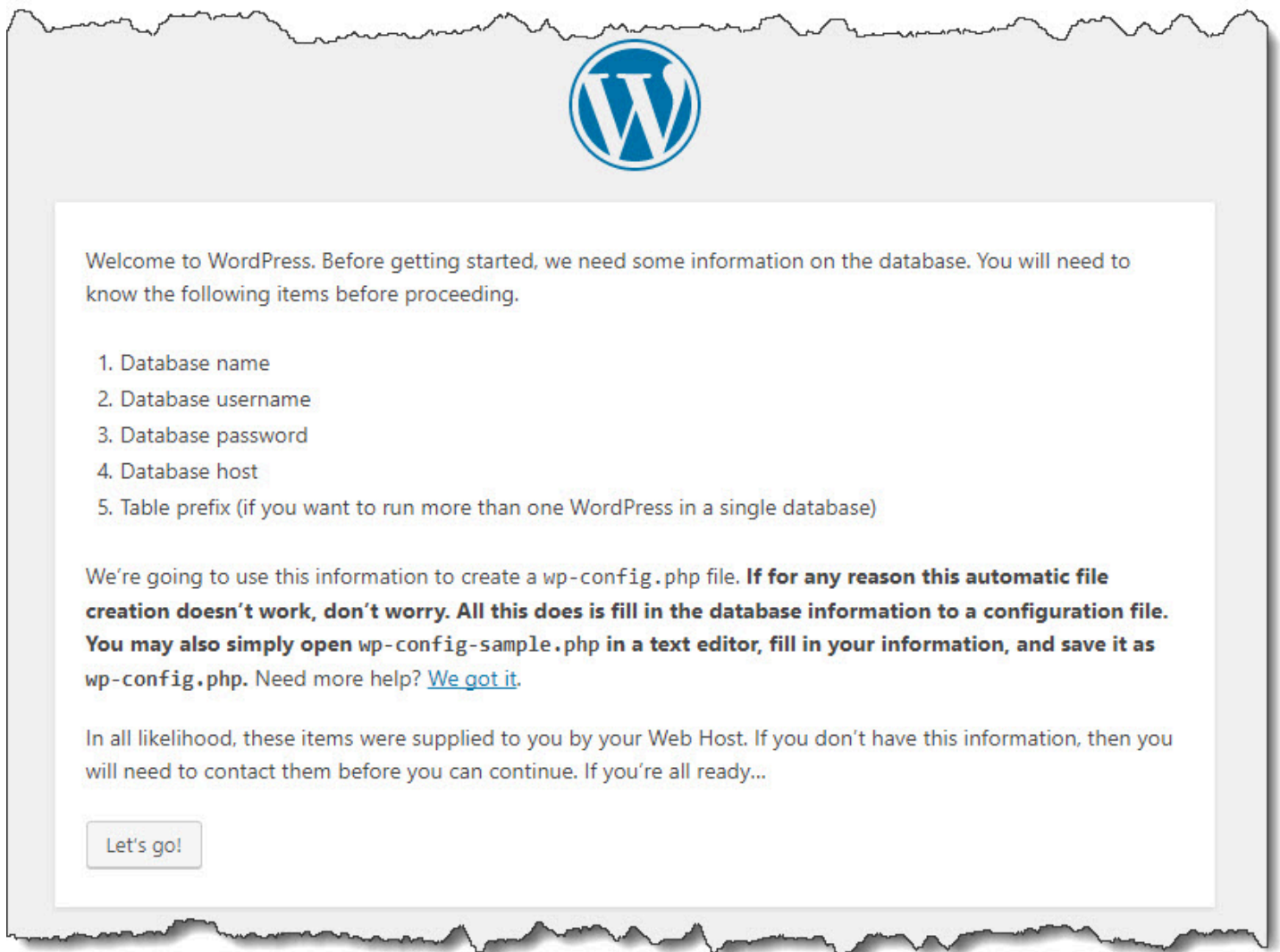
Vérifiez votre déploiement

Une fois le déploiement réussi, vérifiez que votre WordPress installation fonctionne. Utilisez l'adresse DNS publique de l'instance Amazon EC2, suivie de /WordPress, pour consulter votre site dans un navigateur Web. (Pour obtenir la valeur du DNS public, dans la console Amazon EC2, choisissez l'instance Amazon EC2, puis dans l'onglet Description, recherchez la valeur du DNS public.)

Par exemple, si l'adresse DNS publique de votre instance Amazon EC2 est **ec2-01-234-567-890.compute-1.amazonaws.com**, vous devez utiliser l'URL suivante :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress
```

Lorsque vous consultez le site dans votre navigateur, vous devriez voir une page d' WordPress accueil semblable à la suivante :



Si aucune règle HTTP entrante n'est ajoutée à votre instance Amazon EC2 dans son groupe de sécurité, la page d'accueil de WordPress ne s'affiche pas. Si vous voyez un message indiquant que le serveur distant ne répond pas, assurez-vous que le groupe de sécurité de votre instance Amazon EC2 dispose de la règle d'entrée. Pour de plus amples informations, veuillez consulter [Ajoutez une règle entrante qui autorise le trafic HTTP vers votre instance Amazon Linux ou RHEL Amazon EC2](#).

Étape 5 : Mettez à jour et redéployez votre application WordPress

Maintenant que vous avez déployé avec succès la révision de votre application, mettez à jour le WordPress code sur la machine de développement, puis utilisez-le CodeDeploy pour redéployer le site. Ensuite, vous devriez voir les modifications de code sur l'instance Amazon EC2.

Rubriques

- [Configurez le WordPress site](#)
- [Modifier le site](#)
- [Redéployer le site](#)

Configurez le WordPress site

Pour voir les effets de la modification du code, terminez la configuration du WordPress site afin de disposer d'une installation entièrement fonctionnelle.

1. Tapez l'URL de votre site dans votre navigateur Web. L'URL est l'adresse DNS publique de l'instance Amazon EC2 plus une /WordPress extension. Pour cet exemple de WordPress site (et par exemple l'adresse DNS publique de l'instance Amazon EC2), l'URL est. **http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress**
2. Si vous n'avez pas encore configuré le site, la page d'accueil WordPress par défaut s'affiche. Choisissez C'est parti !.
3. Pour utiliser la base de données MySQL par défaut, sur la page de configuration de base de données, tapez les valeurs suivantes :
 - Nom de base de données : **test**
 - Nom d'utilisateur : **root**
 - Mot de passe : laissez vide.
 - Hôte de base de données : **localhost**
 - Préfixe de tableau : **wp_**

Choisissez Soumettre pour configurer la base de données.

4. Poursuivez la configuration du site. Sur la page de bienvenue, saisissez les valeurs souhaitées, puis choisissez Installer WordPress. Une fois l'installation terminée, vous pouvez vous connecter à votre tableau de bord.

Important

Pendant le déploiement de l' WordPress application, le **change_permissions.sh** script a mis à jour les autorisations du /tmp/WordPress dossier afin que tout le monde puisse y écrire. C'est le bon moment pour exécuter la commande suivante afin de limiter les

autorisations de manière à être le seul, en tant que propriétaire, à pouvoir écrire dans celui-ci :

```
chmod -R 755 /var/www/html/WordPress
```

Modifier le site

Pour modifier le WordPress site, rendez-vous dans le dossier de l'application sur votre machine de développement :

```
cd /tmp/WordPress
```

Pour modifier certaines couleurs du site, dans le fichier `wp-content/themes/twentyfifteen/style.css`, utilisez un éditeur de texte ou `sed` pour remplacer `#fff` par `#768331`.

Sous Linux ou d'autres systèmes avec GNU `sed`, utilisez :

```
sed -i 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

Sous macOS, Unix ou d'autres systèmes avec BSD `sed`, utilisez :

```
sed -i '' 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

Redéployer le site

Maintenant que vous avez modifié le code du site, utilisez Amazon S3 CodeDeploy pour redéployer le site.

Regroupez et chargez les modifications sur Amazon S3, comme décrit dans [Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive](#). (Lorsque vous suivez ces instructions, souvenez-vous que vous n'avez pas besoin de créer une application.) Donnez à la nouvelle révision la même clé qu'auparavant (**WordPressApp.zip**). Téléchargez-le dans le même compartiment Amazon S3 que vous avez créé précédemment (par exemple, **amzn-s3-demo-bucket**).

Utilisez le AWS CLI, la CodeDeploy console ou le CodeDeploy APIs pour redéployer le site.

Rubriques

- [Pour redéployer le site \(interface CLI\)](#)
- [Pour redéployer le site \(console\)](#)

Pour redéployer le site (interface CLI)

Appelez la commande `create-deployment` pour créer un déploiement basé sur la révision nouvellement chargée. Utilisez l'application nommée **WordPress_App**, la configuration de déploiement nommée **CodeDeployDefault.OneAtATime**, le groupe de déploiement nommé **WordPress_DepGroup** et la révision nommée **WordPressApp.zip** dans le compartiment nommé **amzn-s3-demo-bucket** :

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DepGroup \  
  --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=WordPressApp.zip
```

Vous pouvez vérifier l'état du déploiement, comme décrit dans la rubrique [Surveillez et résolvez les problèmes liés à votre déploiement](#).

Après avoir CodeDeploy redéployé le site, revisitez-le dans votre navigateur Web pour vérifier que les couleurs ont été modifiées. (Vous devrez peut-être actualiser votre navigateur.) Si les couleurs ont été modifiées, vous avez réussi la procédure. Vous avez réussi à modifier et à redéployer votre site.

Pour redéployer le site (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez **Deploy**, puis choisissez **Applications**.
3. Dans la liste des applications, choisissez **WordPress_App**.
4. Sur l'onglet **Groupe de déploiement**, choisissez **WordPress_DepGroup**.

5. Choisissez Créer un déploiement.
6. Sur la page Créer un déploiement :
 - a. Dans Groupe de déploiement, choisissez **WordPress_DepGroup**.
 - b. Dans la zone Type de référentiel, choisissez Mon application est stockée dans Amazon S3, puis copiez le lien Amazon S3 de votre révision dans le champ Emplacement de la révision. Pour trouver la valeur du lien :
 - i. Dans un onglet de navigateur distinct :

Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/s3/>.

Accédez à amzn-s3-demo-bucket et ouvrez-le, puis choisissez votre révision, **WordPressApp.zip**
 - ii. Si le volet Propriétés n'est pas visible dans la console Amazon S3, cliquez sur le bouton Propriétés.
 - iii. Dans le volet Propriétés, copiez la valeur du champ Lien dans le champ Emplacement des révisions de la CodeDeploy console.
 - c. Si un message s'affiche et vous informe que le type de fichier n'a pas pu être détecté, choisissez .zip.
 - d. Laissez la zone Deployment description vide.
 - e. Développez les remplacements des groupes de déploiement et choisissez CodeDeployDefault dans Configuration du déploiement. OneAtATime.
 - f. Choisissez Démarrer le déploiement. Des informations sur votre déploiement nouvellement créé s'affichent dans la page Deployments.
 - g. Vous pouvez vérifier l'état du déploiement, comme décrit dans la rubrique [Surveillez et résolvez les problèmes liés à votre déploiement](#).

Après avoir CodeDeploy redéployé le site, revisitez-le dans votre navigateur Web pour vérifier que les couleurs ont été modifiées. (Vous devrez peut-être actualiser votre navigateur.) Si les couleurs ont été modifiées, vous avez réussi la procédure. Vous avez réussi à modifier et à redéployer votre site.

Étape 6 : Nettoyez votre WordPress application et les ressources associées

Vous avez maintenant correctement mis à jour le WordPress code et redéployé le site. Pour éviter des frais permanents pour les ressources que vous avez créées pour ce didacticiel, vous devez supprimer :

- Toutes les CloudFormation piles (ou résiliez toute instance Amazon EC2, si vous les avez créées en dehors CloudFormation de).
- Tous les compartiments Amazon S3.
- l'application WordPress_App dans CodeDeploy ;
- L'association AWS Systems Manager State Manager de l' CodeDeploy agent.

Vous pouvez utiliser les AWS CLI consoles Amazon S3, Amazon EC2 et les CodeDeploy consoles, ou AWS APIs pour effectuer le nettoyage. CloudFormation

Rubriques

- [Pour nettoyer les ressources \(interface CLI\)](#)
- [Pour nettoyer les ressources \(console\)](#)
- [Quelle est la prochaine étape ?](#)

Pour nettoyer les ressources (interface CLI)

1. Si vous avez utilisé notre CloudFormation modèle pour ce didacticiel, appelez la delete-stack commande par rapport à la pile nommée **CodeDeployDemoStack**. Cela mettra fin à toutes les instances Amazon EC2 associées et supprimera tous les rôles IAM associés créés par la pile :

```
aws cloudformation delete-stack --stack-name CodeDeployDemoStack
```

2. Pour supprimer le compartiment Amazon S3, appelez la rm commande en utilisant le --recursive commutateur correspondant au compartiment nommé **amzn-s3-demo-bucket**. Elle supprime le compartiment et tous les objets qui y figurent :

```
aws s3 rm s3://amzn-s3-demo-bucket --recursive --region region
```

3. Pour supprimer l'application WordPress_App, appelez la commande delete-application. Elle supprime également tous les enregistrements de groupe de déploiement et les enregistrements de déploiement associés pour l'application :

```
aws deploy delete-application --application-name WordPress_App
```

4. Pour supprimer l'association Systems Manager State Manager, appelez la delete-association commande.

```
aws ssm delete-association --association-id association-id
```

Vous pouvez l'obtenir *association-id* en appelant la describe-association commande.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets  
Key=tag:Name,Values=CodeDeployDemo
```

Si vous n'avez pas utilisé la CloudFormation pile pour ce didacticiel, appelez la terminate-instances commande pour mettre fin aux instances Amazon EC2 que vous avez créées manuellement. Fournissez l'ID de l'instance Amazon EC2 à résilier :

```
aws ec2 terminate-instances --instance-ids instanceId
```

Pour nettoyer les ressources (console)

Si vous avez utilisé notre CloudFormation modèle pour ce didacticiel, supprimez la CloudFormation pile associée.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la zone Filtre, tapez le nom de la CloudFormation pile que vous avez créée précédemment (par exemple, **CodeDeployDemoStack**).
3. Cochez la case en regard du nom de la pile. Dans le menu Actions, sélectionnez Supprimer pile.

CloudFormation supprime la pile, met fin à toutes les instances Amazon EC2 associées et supprime tous les rôles IAM associés.

Pour mettre fin aux instances Amazon EC2 que vous avez créées en dehors d'une CloudFormation pile :

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse.
<https://console.aws.amazon.com/ec2/>
2. Dans la liste INSTANCES, sélectionnez Instances.
3. Dans le champ de recherche, tapez le nom de l'instance Amazon EC2 que vous souhaitez résilier (par exemple, **CodeDeployDemo**), puis appuyez sur Entrée.
4. Choisissez le nom de l'instance Amazon EC2.
5. Dans le menu Actions, pointez sur État de l'instance, puis choisissez Terminer. À l'invite, choisissez Oui, résilier.

Répétez ces étapes pour chaque instance.

Pour supprimer le compartiment Amazon S3, procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse
<https://console.aws.amazon.com/s3/>.
2. Dans la liste des compartiments, recherchez et choisissez le nom du compartiment Amazon S3 que vous avez créé précédemment (par exemple, **amzn-s3-demo-bucket**).
3. Avant de pouvoir supprimer un compartiment, vous devez d'abord supprimer son contenu. Sélectionnez tous les fichiers figurant dans le compartiment, tels que **WordPressApp.zip**. Dans le menu Actions, sélectionnez Delete. Lorsque vous êtes invité à confirmer la suppression, choisissez OK.
4. Une fois que le compartiment est vide, vous pouvez le supprimer. Dans la liste des compartiments, sélectionnez la ligne du compartiment (mais pas son nom). Choisissez Supprimer le compartiment, lorsque vous êtes invité à confirmer, puis OK.

Pour supprimer l'WordPress_Application depuis CodeDeploy :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse
<https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.

3. Dans la liste des applications, choisissez WordPress_App.
4. Sur la page Détails de l'application, choisissez Supprimer l'application.
5. À l'invite, entrez le nom de l'application pour confirmer que vous souhaitez la supprimer, puis choisissez Supprimer.

Pour supprimer l'association Systems Manager State Manager, procédez comme suit :

1. Ouvrez la AWS Systems Manager console dans <https://console.aws.amazon.com/systems-manager>.
2. Dans le volet de navigation, sélectionnez State Manager.
3. Sélectionnez l'association que vous avez créée et choisissez Delete (Supprimer).

Quelle est la prochaine étape ?

Félicitations pour être arrivé jusqu'ici ! Vous avez réussi un déploiement CodeDeploy, puis mis à jour le code de votre site et l'avez redéployé.

Tutoriel : Déployez un message « Bonjour tout le monde ! » application avec CodeDeploy (Windows Server)

Dans ce didacticiel, vous allez déployer une page Web unique sur une instance Amazon EC2 Windows Server exécutant Internet Information Services (IIS) comme serveur Web. Cette page Web affiche un simple message « Bonjour tout le monde ! » message.

Ce n'est pas ce que vous recherchez ?

- Pour vous entraîner plutôt à déployer sur une instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux (RHEL), consultez. [Tutoriel : Déploiement WordPress sur une instance Amazon EC2 \(Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix\)](#)
- Pour vous exercer plutôt à déployer sur une instance sur site, consultez la page [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#).

Les étapes de ce didacticiel sont présentées du point de vue de Windows. Bien que vous puissiez effectuer la plupart de ces étapes sur un ordinateur local exécutant Linux, macOS ou Unix, vous

devez adapter celles qui concernent les chemins de répertoire basés sur Windows, tels que. c : \temp En outre, si vous souhaitez vous connecter à l'instance Amazon EC2, vous avez besoin d'une application cliente capable de se connecter via le protocole RDP (Remote Desktop Protocol) à l'instance Amazon EC2 exécutant Windows Server. (Windows inclut une application cliente de connexion RDP par défaut.)

Avant de commencer ce didacticiel, vous devez remplir les conditions requises [Commencer avec CodeDeploy](#), notamment configurer votre utilisateur, installer ou mettre à niveau le AWS CLI, et créer un profil d'instance IAM et un rôle de service.

Rubriques

- [Étape 1 : Lancer une instance Amazon EC2 de Windows Server](#)
- [Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 de Windows Server](#)
- [Étape 3 : Téléchargez votre message « Bonjour tout le monde ! » application pour Amazon S3](#)
- [Étape 4 : Déployez votre application Hello World](#)
- [Étape 5 : Mettez à jour et redéployez votre message « Hello, world ! » vigie](#)
- [Étape 6 : Nettoyez votre message « Bonjour tout le monde ! » application et ressources connexes](#)

Étape 1 : Lancer une instance Amazon EC2 de Windows Server

Pour déployer l'application Hello World avec CodeDeploy, vous avez besoin d'une instance Amazon EC2 exécutant Windows Server.

Suivez les instructions de la section [Créez une instance Amazon EC2 pour CodeDeploy](#). Lorsque vous êtes prêt à attribuer une balise d'instance Amazon EC2 à l'instance, assurez-vous de spécifier la clé de balise **Name** et la valeur de balise de. **CodeDeployDemo** (Si vous spécifiez une clé de balise ou une valeur de balise différente, les instructions fournies dans [Étape 4 : Déployez votre application Hello World](#) peuvent générer des résultats inattendus.)

Après avoir lancé l'instance Amazon EC2, revenez à cette page et passez à la section suivante. Ne passez pas à l'étape [Créez une application avec CodeDeploy](#).

Connectez-vous à votre instance Amazon EC2

Après le lancement de votre instance Amazon EC2, suivez ces instructions pour vous entraîner à vous y connecter.

 Note

Dans ces instructions, nous supposons que vous exécutez Windows et l'application cliente de connexion au Bureau Windows. Pour plus d'informations, consultez [Connexion à votre instance Windows à l'aide du protocole RDP](#). Vous devrez peut-être adapter ces instructions pour d'autres systèmes d'exploitation ou d'autres applications clientes de connexion RDP.

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
2. Dans le panneau de navigation, sous Instances, choisissez Instances.
3. Accédez à votre instance Windows Server et choisissez-la dans la liste.
4. Choisissez Se connecter.
5. Choisissez Get Password (Obtenir un mot de passe), puis choisissez Choose File (Sélectionner un fichier).
6. Accédez au fichier de paire de clés d'instance Amazon EC2 associé à l'instance Amazon EC2 de Windows Server et choisissez-le, puis choisissez Ouvrir.
7. Choisissez Déchiffrer le mot de passe. Notez le mot de passe qui s'affiche. Vous en aurez besoin à l'étape 10.
8. Choisissez Télécharger le fichier Bureau à distance, puis ouvrez le fichier.
9. Si vous êtes invité à vous connecter alors que l'éditeur de la connexion à distance ne peut pas être identifié, continuez.
10. Saisissez le mot de passe que vous avez noté à l'étape 7, puis continuez. (Si votre application cliente de connexion RDP vous invite à fournir un nom utilisateur, tapez **Administrator**.)
11. Si vous êtes invité à vous connecter alors que l'identité de l'ordinateur distant ne peut pas être vérifiée, continuez.
12. Une fois connecté, le bureau de l'instance Amazon EC2 exécutant Windows Server s'affiche.
13. Vous pouvez désormais vous déconnecter de l'instance Amazon EC2.

 Warning

N'arrêtez pas ni ne mettez hors service l'instance. Sinon, CodeDeploy impossible de le déployer.

Ajoutez une règle entrante qui autorise le trafic HTTP vers votre instance Amazon EC2 de Windows Server

L'étape suivante confirme que votre instance Amazon EC2 possède un port HTTP ouvert afin que vous puissiez voir la page Web déployée sur votre instance Amazon EC2 Windows Server dans un navigateur.

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
2. Choisissez Instances, puis choisissez votre instance.
3. Dans l'onglet Description, sous Groupes de sécurité, choisissez Afficher les règles entrantes.

Vous devriez voir une liste de règles dans votre groupe de sécurité, comme suit :

```
Security Groups associated with i-1234567890abcdef0
Ports      Protocol    Source      launch-wizard-N
22         tcp        0.0.0.0/0   #
```

4. Sous Groupes de sécurité, choisissez le groupe de sécurité pour votre instance Amazon EC2. Il pourrait être nommé **launch-wizard-*N***. Le ***N*** dans le nom est un numéro attribué à votre groupe de sécurité lorsque l'instance a été créée.

Choisissez l'onglet Entrant. Si le groupe de sécurité de votre instance est correctement configuré, vous devriez voir apparaître une règle avec les valeurs suivantes :

- Type : HTTP
 - Protocole : TCP
 - Plage de ports : 80
 - Source : 0.0.0.0/0
5. Si aucune règle ne contient ces valeurs, utilisez les procédures décrites dans [Ajouter des règles à un groupe de sécurité](#) pour les ajouter à une nouvelle règle de sécurité.

Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 de Windows Server

Il est maintenant temps de configurer le contenu source de votre application afin de pouvoir le déployer sur l'instance Amazon EC2. Dans le cadre de ce didacticiel, vous allez déployer une seule

page Web sur l'instance Amazon EC2 exécutant Windows Server, qui exécutera Internet Information Services (IIS) en tant que serveur Web. Cette page Web affichera un simple message « Bonjour tout le monde ! » message.

Rubriques

- [Création de la page Web](#)
- [Créez un script pour exécuter votre application](#)
- [Ajouter un fichier de spécification d'application](#)

Création de la page Web

1. Créez un sous-répertoire (sous-dossier) nommé HelloWorldApp dans votre dossier c:\temp, puis basculez vers ce dossier.

```
mkdir c:\temp\HelloWorldApp
cd c:\temp\HelloWorldApp
```

Note

Vous n'êtes pas tenu d'utiliser l'emplacement c:\temp ni le nom de sous-dossier HelloWorldApp. Si vous utilisez un autre emplacement ou nom de sous-dossier, veillez à l'utiliser tout au long de ce didacticiel.

2. Utilisez un éditeur de texte pour créer un fichier dans le dossier. Nommez le fichier index.html.

```
notepad index.html
```

3. Ajoutez le code HTML suivant au fichier, puis enregistrez le fichier.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Hello, World!</title>
  <style>
    body {
      color: #ffffff;
```

```
        background-color: #0188cc;
        font-family: Arial, sans-serif;
        font-size:14px;
    }
</style>
</head>
<body>
    <div align="center"><h1>Hello, World!</h1></div>
    <div align="center"><h2>You have successfully deployed an application using
CodeDeploy</h2></div>
    <div align="center">
        <p>What to do next? Take a look through the <a href="https://aws.amazon.com/
codedeploy">CodeDeploy Documentation</a>.</p>
    </div>
</body>
</html>
```

Créez un script pour exécuter votre application

Vous allez ensuite créer un script que CodeDeploy utilisera pour configurer le serveur Web sur l'instance Amazon EC2 cible.

1. Dans le sous-dossier où le fichier `index.html` est enregistré, utilisez un éditeur de texte pour créer un autre fichier. Nommez le fichier `before-install.bat`.

```
notepad before-install.bat
```

2. Ajoutez le code de script de commandes par lot suivant dans le fichier, puis enregistrez le fichier.

```
REM Install Internet Information Server (IIS).
c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Import-Module -
Name ServerManager
c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Install-
WindowsFeature Web-Server
```

Ajouter un fichier de spécification d'application

Vous allez ensuite ajouter un fichier de spécification d'application (AppSpec fichier) en plus de la page Web et du fichier de script batch. Le AppSpec fichier est un fichier au format [YAML](#) utilisé pour : CodeDeploy

- mapper les fichiers sources figurant dans votre révision d'application vers leurs destinations sur l'instance ;
- spécifier des scripts à exécuter sur l'instance au cours du déploiement.

Le AppSpec fichier doit être nommé `appspec.yml`. Il doit être placé dans le dossier racine du code source de l'application.

1. Dans le sous-dossier où les fichiers `index.html` et `before-install.bat` sont enregistrés, utilisez un éditeur de texte pour créer un autre fichier. Nommez le fichier `appspec.yml`.

```
notepad appspec.yml
```

2. Ajoutez le code YAML suivant au fichier, puis enregistrez le fichier.

```
version: 0.0
os: windows
files:
  - source: \index.html
    destination: c:\inetpub\wwwroot
hooks:
  BeforeInstall:
    - location: \before-install.bat
      timeout: 900
```

CodeDeploy utilisera ce AppSpec fichier pour copier le `index.html` fichier du dossier racine du code source de l'application vers le `c:\inetpub\wwwroot` dossier de l'instance Amazon EC2 cible. Pendant le déploiement, CodeDeploy exécutera le script `before-install.bat` batch sur l'instance Amazon EC2 cible pendant l'événement du cycle de vie du **BeforeInstall** déploiement. Si l'exécution de ce script prend plus de 900 secondes (15 minutes), il CodeDeploy arrêtera le déploiement et marquera le déploiement sur l'instance Amazon EC2 comme ayant échoué.

Pour plus d'informations sur ces paramètres, consultez la rubrique [CodeDeploy AppSpec référence de fichier](#).

Important

Les emplacements et les nombres d'espaces entre les différents éléments de ce fichier sont importants. Si l'espacement est incorrect, cela CodeDeploy provoquera une erreur qui

peut être difficile à déboguer. Pour de plus amples informations, veuillez consulter [AppSpec](#) [Espace des fichiers](#).

Étape 3 : Téléchargez votre message « Bonjour tout le monde ! » application pour Amazon S3

Vous allez maintenant préparer et télécharger votre contenu source vers un emplacement à partir duquel CodeDeploy vous pourrez le déployer. Les instructions suivantes vous montrent comment configurer un compartiment Amazon S3, préparer les fichiers de révision de l'application pour le compartiment, regrouper les fichiers de révision, puis transférer la révision dans le compartiment.

Note

Bien que cela ne soit pas abordé dans ce didacticiel, vous pouvez l'utiliser CodeDeploy pour déployer des applications depuis des GitHub référentiels vers des instances. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy avec GitHub](#).

Rubriques

- [Provisionner un compartiment Amazon S3](#)
- [Préparer les fichiers de l'application pour le bucket](#)
- [Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive](#)

Provisionner un compartiment Amazon S3

Créez un conteneur ou un compartiment de stockage dans Amazon S3 ou utilisez un compartiment existant. Assurez-vous que vous pouvez télécharger la révision dans le compartiment et que les instances Amazon EC2 utilisées dans les déploiements peuvent télécharger la révision depuis le compartiment.

Vous pouvez utiliser AWS CLI la console Amazon S3 ou Amazon S3 APIs pour créer un compartiment Amazon S3. Une fois que vous avez créé le compartiment, veillez à accorder des autorisations d'accès au compartiment et à votre utilisateur CodeDeploy.

 Note

Les noms de compartiment doivent être uniques sur Amazon S3 pour tous les AWS comptes. Si vous n'êtes pas en mesure d'utiliser **amzn-s3-demo-bucket**, essayez un autre nom de compartiment, tel que **amzn-s3-demo-bucket**, suivi d'un tiret et de vos initiales ou d'un autre identificateur unique. Veillez ensuite à substituer votre nom de compartiment à **amzn-s3-demo-bucket** chaque fois que vous le verrez tout au long de ce didacticiel.

Le compartiment Amazon S3 doit être créé dans la même AWS région que celle dans laquelle vos instances Amazon EC2 cibles sont lancées. Par exemple, si vous créez le bucket dans la région USA Est (Virginie du Nord), vos instances Amazon EC2 cibles doivent être lancées dans la région USA Est (Virginie du Nord).

Rubriques

- [Pour créer un compartiment Amazon S3 \(CLI\)](#)
- [Pour créer un compartiment Amazon S3 \(console\)](#)
- [Donnez des autorisations au compartiment Amazon S3 et à votre AWS compte](#)

Pour créer un compartiment Amazon S3 (CLI)

Appelez la mb commande pour créer un compartiment Amazon S3 nommé **amzn-s3-demo-bucket** :

```
aws s3 mb s3://amzn-s3-demo-bucket --region region
```

Pour créer un compartiment Amazon S3 (console)

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Dans la console Amazon S3, choisissez Create bucket.
3. Dans la zone Nom du compartiment, saisissez un nom pour le compartiment.
4. Dans la liste Région, choisissez la région cible, puis Créer.

Donnez des autorisations au compartiment Amazon S3 et à votre AWS compte

Vous devez disposer des autorisations nécessaires pour effectuer le téléchargement dans le compartiment Amazon S3. Vous pouvez spécifier ces autorisations par le biais d'une politique de

compartiment Amazon S3. Par exemple, dans la politique de compartiment Amazon S3 suivante, l'utilisation du caractère générique (*) permet 111122223333 au AWS compte de télécharger des fichiers dans n'importe quel répertoire du compartiment Amazon S3 nommé amzn-s3-demo-bucket :

```
{
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      }
    }
  ]
}
```

Pour consulter l'identifiant de votre AWS compte, consultez la section [Trouver votre identifiant de AWS compte](#).

C'est le bon moment pour vérifier que le compartiment Amazon S3 autorisera les demandes de téléchargement provenant de chaque instance Amazon EC2 participante. Vous pouvez le spécifier par le biais d'une politique de compartiment Amazon S3. Par exemple, dans la politique de compartiment Amazon S3 suivante, l'utilisation du caractère générique (*) permet à toute instance Amazon EC2 associée à un profil d'instance IAM contenant l'arn:aws:iam::444455556666:role/CodeDeployDemoARN de télécharger des fichiers depuis n'importe quel répertoire du compartiment Amazon S3 nommé : amzn-s3-demo-bucket

```
{
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
```

```

    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Principal": {
      "AWS": [
        "arn:aws:iam::444455556666:role/CodeDeployDemo"
      ]
    }
  ]
}

```

Pour plus d'informations sur la façon de générer et d'associer une politique de compartiment Amazon S3, consultez les [exemples de politique de compartiment](#).

L'utilisateur CodeDeploy administratif que vous avez créé [Étape 1 : Configuration](#) doit également être autorisé à télécharger la révision dans le compartiment Amazon S3. Pour le spécifier, vous pouvez notamment utiliser une stratégie IAM, que vous ajoutez à l'ensemble d'autorisations de l'utilisateur, ou un rôle IAM (que vous autorisez l'utilisateur à assumer). La politique IAM suivante permet à l'utilisateur de télécharger des révisions n'importe où dans le compartiment Amazon S3 nommé `amzn-s3-demo-bucket` :

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}

```

Pour plus d'informations sur la création d'une stratégie IAM, consultez la section [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'ajout d'une politique à un ensemble d'autorisations, voir [Création d'un ensemble d'autorisations](#) dans le Guide de AWS IAM Identity Center l'utilisateur.

Préparer les fichiers de l'application pour le bucket

Assurez-vous que la page Web, le AppSpec fichier et le script sont organisés sur votre machine de développement comme suit :

```
c:\
|-- temp\
    |-- HelloWorldApp\
        |-- appspec.yml
        |-- before-install.bat
        |-- index.html
```

Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive

Groupez les fichiers en un fichier d'archive (appelé révision d'application).

Note

Il est possible qu'il vous soit facturé le stockage des objets dans un compartiment et le transfert des révisions d'application vers et hors d'un compartiment. Pour plus d'informations, consultez [Tarification Amazon S3](#).

1. Sur la machine de développement, basculez vers le dossier dans lequel les fichiers sont stockés :

```
cd c:\temp\HelloWorldApp
```

Note

Si vous ne basculez pas dans ce dossier, la création d'un bundle de fichiers commencera dans votre dossier actif. Par exemple, si votre dossier actif est `c:\temp` au lieu de `c:\temp\HelloWorldApp`, la création du bundle commencera avec les fichiers et les sous-dossiers figurant dans le dossier `c:\temp`, lequel peut inclure plus que le simple sous-dossier `HelloWorldApp`.

2. Appelez la create-application commande pour enregistrer une nouvelle application nommée **HelloWorld_App** avec CodeDeploy :


```
aws deploy create-application --application-name HelloWorld_App
```

3. Appelez la commande CodeDeploy [push](#) pour regrouper les fichiers, télécharger les révisions CodeDeploy sur Amazon S3 et enregistrer les informations relatives à la révision téléchargée, le tout en une seule action.

```
aws deploy push --application-name HelloWorld_App --s3-location s3://amzn-s3-demo-bucket/HelloWorld_App.zip --ignore-hidden-files
```

Cette commande regroupe les fichiers du répertoire actuel (à l'exception des fichiers cachés) dans un seul fichier d'archive nommé `HelloWorld_App.zip`, télécharge la révision dans le **amzn-s3-demo-bucket** compartiment et enregistre les informations relatives à CodeDeploy la révision téléchargée.

Étape 4 : Déployez votre application Hello World

Vous devez maintenant déployer l'exemple de révision de l'application Hello World que vous avez chargé sur Amazon S3. Vous utilisez la console AWS CLI ou la CodeDeploy console pour déployer la révision et suivre la progression du déploiement. Une fois que la révision d'application est déployée avec succès, vous vérifiez les résultats.

Rubriques

- [Déployez la révision de votre application avec CodeDeploy](#)
- [Surveillez et résolvez les problèmes liés à votre déploiement](#)
- [Vérifiez votre déploiement](#)

Déployez la révision de votre application avec CodeDeploy

Vous pouvez déployer votre application via l'interface de ligne de commande ou la console.

Rubriques

- [Pour déployer votre révision d'application \(interface CLI\)](#)
- [Pour déployer votre révision d'application \(console\)](#)

Pour déployer votre révision d'application (interface CLI)

1. Tout d'abord, le déploiement a besoin d'un groupe de déploiement. Toutefois, avant de créer le groupe de déploiement, vous avez besoin d'un nom ARN de rôle de service. Un rôle de service est un rôle IAM qui autorise un service à agir en votre nom. Dans ce cas, le rôle de service donne CodeDeploy l'autorisation d'accéder à vos instances Amazon EC2 pour étendre (lire) leurs balises d'instance Amazon EC2.

Vous devriez avoir déjà suivi les instructions de la rubrique [Création d'un rôle de service \(CLI\)](#) pour créer un rôle de service. Pour obtenir le nom ARN du rôle de service, consultez la page [Obtenez l'ARN du rôle de service \(CLI\)](#).

2. Maintenant que vous avez l'ARN, appelez la `create-deployment-group` commande pour créer un groupe de déploiement nommé **HelloWorld_DepGroup**, associé à l'application nommée **HelloWorld_App**, à l'aide de la balise d'instance Amazon EC2 **CodeDeployDemo** et du nom de la configuration de déploiement **CodeDeployDefault.OneAtATime**, avec l'ARN du rôle de service :

```
aws deploy create-deployment-group --application-name HelloWorld_App
--deployment-group-name HelloWorld_DepGroup --deployment-
config-name CodeDeployDefault.OneAtATime --ec2-tag-filters
Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE --service-role-arn serviceRoleARN
```

Note

La [create-deployment-group](#) commande permet de créer des déclencheurs qui entraînent l'envoi de notifications Amazon SNS aux abonnés du topic concernant des événements spécifiques liés aux déploiements et aux instances. La commande prend également en charge les options permettant d'annuler automatiquement les déploiements et de configurer des alarmes pour arrêter les déploiements lorsque les seuils de surveillance des CloudWatch alarmes d'Amazon sont atteints. Les commandes associées à ces actions ne sont pas incluses dans ce didacticiel.

3. Avant de créer un déploiement, l' `CodeDeploy` agent doit être installé sur les instances de votre groupe de déploiement. Vous pouvez installer l'agent à partir de la ligne de commande AWS Systems Manager avec la commande suivante :

```
aws ssm create-association --name AWS-ConfigureAWSPackage
--targets Key=tag:Name,Values=CodeDeployDemo --parameters
```

```
action=Install,name=AWSCodeDeployAgent --schedule-expression "cron(0 2 ? * SUN
*)"
```

Cette commande crée une association dans Systems Manager State Manager qui installera l'CodeDeploy agent puis tentera de le mettre à jour à 2h00 tous les dimanches matins. Pour plus d'informations sur l'CodeDeploy agent, consultez la section [Utilisation de l'CodeDeploy agent](#). Pour plus d'informations sur Systems Manager, consultez [What is AWS Systems Manager](#).

4. Maintenant, appelez la commande `create-deployment` pour créer un déploiement associé à l'application nommée **HelloWorld_App**, à la configuration de déploiement nommée **CodeDeployDefault.OneAtATime** et au groupe de déploiement nommé **HelloWorld_DepGroup**, en utilisant la révision d'application nommée **HelloWorld_App.zip** dans le compartiment nommé **amzn-s3-demo-bucket** :

```
aws deploy create-deployment --application-name HelloWorld_App --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
```

Pour déployer votre révision d'application (console)

1. Avant d'utiliser la CodeDeploy console pour déployer la révision de votre application, vous avez besoin d'un ARN de rôle de service. Un rôle de service est un rôle IAM qui autorise un service à agir en votre nom. Dans ce cas, le rôle de service donne CodeDeploy l'autorisation d'accéder à vos instances Amazon EC2 pour étendre (lire) leurs balises d'instance Amazon EC2.

Vous devriez avoir déjà suivi les instructions de la rubrique [Création d'un rôle de service \(console\)](#) pour créer un rôle de service. Pour obtenir le nom ARN du rôle de service, consultez la page [Obtenir l'ARN du rôle de service \(console\)](#).

2. Maintenant que vous disposez de l'ARN, vous pouvez utiliser la CodeDeploy console pour déployer la révision de votre application.

Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

3. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
4. Choisissez HelloWorld_App.
5. Dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement.
6. Dans Nom du groupe de déploiement, saisissez **HelloWorld_DepGroup**.
7. Dans Rôle de service, choisissez le nom du rôle de service.
8. Dans Type de déploiement, choisissez Sur place.
9. Dans Configuration de l'environnement, sélectionnez les instances Amazon EC2.
10. Dans la configuration de l'agent avec AWS Systems Manager, conservez les valeurs par défaut.
11. Dans Clé, entrez **Name**.
12. Dans Value (Valeur), entrez **CodeDeployDemo**.
13. Dans Configuration du déploiement, choisissez CodeDeployDefault. OneAtATime.
14. Dans Équilibreur de charge, désactivez Activer l'équilibrage de charge.
15. Choisissez Créer un groupe de déploiement.
16. Choisissez Créer un déploiement.
17. Dans le groupe de déploiement, choisissez HelloWorld_ DepGroup
18. Dans Type de révision, choisissez Mon application est stockée dans Amazon S3, puis dans Emplacement de la révision, entrez l'emplacement de l'exemple de révision de l'application Hello World que vous avez précédemment chargé sur Amazon S3. Pour obtenir cet emplacement :
 - a. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
 - b. Dans la liste des buckets, choisissez amzn-s3-demo-bucket (ou le nom du bucket dans lequel vous avez chargé la révision de votre application).
 - c. Dans la liste des objets, choisissez HelloWorld_App.zip.
 - d. Dans l'onglet Overview (Présentation), choisissez Copy path (Copier le chemin).
 - e. Retournez à la CodeDeploy console, puis dans Emplacement des révisions, collez la valeur du champ Lien.
19. Pour Revision file type (Type de fichier de révision), choisissez .zip.
20. (Facultatif) Entrez un commentaire dans Deployment description) Description du déploiement.
21. Choisissez Créer un déploiement. Des informations sur votre déploiement nouvellement créé s'affichent dans la page Deployments.

Surveillez et résolvez les problèmes liés à votre déploiement

Utilisez la console AWS CLI ou la console pour surveiller et résoudre les problèmes liés à votre déploiement.

Rubriques

- [Pour surveiller et déboguer votre déploiement \(interface CLI\)](#)
- [Pour surveiller et déboguer votre déploiement \(console\)](#)

Pour surveiller et déboguer votre déploiement (interface CLI)

1. Obtenez l'ID du déploiement en appelant la commande `list-deployments` sur l'application nommée **HelloWorld_App** et le groupe de déploiement nommé **HelloWorld_DepGroup** :

```
aws deploy list-deployments --application-name HelloWorld_App --deployment-group-name HelloWorld_DepGroup --query "deployments" --output text
```

2. Appelez la commande `get-deployment` avec l'ID du déploiement :

```
aws deploy get-deployment --deployment-id deploymentID --query "deploymentInfo.status" --output text
```

3. La commande renvoie l'état général du déploiement. En cas de réussite, la valeur est `Succeeded`.

Si l'état général est `Failed`, vous pouvez appeler des commandes telles que [list-deployment-instances](#) et [get-deployment-instance](#) pour résoudre les problèmes. Pour plus d'options de débogage, consultez la page [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#).

Pour surveiller et déboguer votre déploiement (console)

Sur la page Déploiements de la CodeDeploy console, vous pouvez surveiller l'état de votre déploiement dans la colonne État.

Pour obtenir plus d'informations sur votre déploiement, notamment si la valeur de la colonne État a une valeur différente de `Succeeded` :

1. Dans la table Déploiements, choisissez votre ID de déploiement. Une fois qu'un déploiement a échoué, un message décrivant la raison de l'échec s'affiche sur la page des détails du déploiement.
2. Des informations supplémentaires sur les instances du déploiement sont affichées. Après l'échec d'un déploiement, vous pourrez peut-être déterminer sur quelles instances Amazon EC2 et à quelle étape le déploiement a échoué.
3. Si vous souhaitez effectuer davantage d'opérations de dépannage, vous pouvez utiliser une technique telle que celle exposée dans la rubrique [View Instance Details](#). Vous pouvez également analyser les fichiers journaux de déploiement sur une instance Amazon EC2. Pour de plus amples informations, veuillez consulter [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#).

Vérifiez votre déploiement

Une fois que votre déploiement a réussi, vérifiez que votre installation fonctionne. Utilisez l'adresse DNS publique de l'instance Amazon EC2 pour afficher la page Web dans un navigateur Web. (Pour obtenir la valeur du DNS public, dans la console Amazon EC2, choisissez l'instance Amazon EC2, puis dans l'onglet Description, recherchez la valeur dans Public DNS.)

Par exemple, si l'adresse DNS publique de votre instance Amazon EC2 est **ec2-01-234-567-890.compute-1.amazonaws.com**, vous devez utiliser l'URL suivante :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

En cas de succès, vous devez voir une page Web « Hello, World! ».

Étape 5 : Mettez à jour et redéployez votre message « Hello, world ! » vigie

Maintenant que vous avez déployé avec succès la révision de votre application, sur la machine de développement, mettez à jour le code de la page Web, puis utilisez-le CodeDeploy pour redéployer le site. Après le redéploiement, vous devriez être en mesure de voir les modifications sur l'instance Amazon EC2.

Rubriques

- [Modifier la page Web](#)
- [Redéployer le site](#)

Modifier la page Web

1. Accédez à votre sous-dossier `c:\temp\HelloWorldApp` et utilisez un éditeur de texte pour modifier le fichier `index.html` :

```
cd c:\temp\HelloWorldApp
notepad index.html
```

2. Révisez le contenu du fichier `index.html` pour modifier la couleur d'arrière-plan et une partie du texte sur la page web, puis enregistrez le fichier :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Hello Again, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #66cc00;
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello Again, World!</h1></div>
  <div align="center"><h2>You have successfully deployed a revision of an
application using CodeDeploy</h2></div>
  <div align="center">
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Redéployer le site

Maintenant que vous avez modifié le code, utilisez Amazon S3 CodeDeploy pour redéployer la page Web.

Regroupez et chargez les modifications sur Amazon S3 comme décrit dans [Regroupez les fichiers de l'application dans un seul fichier d'archive et transférez le fichier d'archive](#). (Lorsque vous suivez ces instructions, vous n'avez pas besoin de créer une nouvelle application.) Donnez à la révision la même clé qu'auparavant (**HelloWorld_App.zip**). Téléchargez-le dans le même compartiment Amazon S3 que vous avez créé précédemment (par exemple, **amzn-s3-demo-bucket**).

Utilisez la console AWS CLI ou la CodeDeploy console pour redéployer le site.

Rubriques

- [Pour redéployer le site \(interface CLI\)](#)
- [Pour redéployer le site \(console\)](#)

Pour redéployer le site (interface CLI)

Appelez la commande `create-deployment` pour créer un déploiement basé sur la révision téléchargée, de nouveau en utilisant l'application nommée **HelloWorld_App**, la configuration de déploiement nommée **CodeDeployDefault.OneAtATime** et le groupe de déploiement nommé **HelloWorld_DepGroup** et la révision nommée **HelloWorld_App.zip** dans le compartiment nommé **amzn-s3-demo-bucket** :

```
aws deploy create-deployment --application-name HelloWorld_App --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-location bucket=amzn-s3-demo-bucket,bundleType=zip,key=HelloWorld_App.zip
```

Vous pouvez vérifier l'état du nouveau déploiement, comme décrit dans la rubrique [Surveillez et résolvez les problèmes liés à votre déploiement](#).

Une fois le site CodeDeploy redéployé, revisitez-le dans votre navigateur Web pour vérifier que la couleur d'arrière-plan et le texte de la page Web ont été modifiés. (Vous devrez peut-être actualiser votre navigateur.) Si la couleur d'arrière-plan et le texte ont été modifiés, vous avez réussi la procédure. Vous avez modifié et redéployé votre site.

Pour redéployer le site (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation , choisissez Applications.
3. Dans la liste des applications, choisissez HelloWorld_App.
4. Dans l'onglet Déploiements, choisissez Créer un déploiement.
 - a. Dans la liste des groupes de déploiement, choisissez HelloWorld_ DepGroup.
 - b. Dans Emplacement de la révision, entrez le lien Amazon S3 correspondant à votre révision.

Pour trouver la valeur du lien :

- i. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/s3/>.

Accédez à amzn-s3-demo-bucket et ouvrez-le, puis choisissez votre révision dans **HelloWorld_App.zip** la console Amazon S3.

- ii. Si le volet Propriétés n'est pas visible dans la console Amazon S3, cliquez sur le bouton Propriétés.
- iii. Dans le volet Propriétés, copiez la valeur du champ Lien.
- iv. Retournez à la CodeDeploy console, puis collez le lien dans l'emplacement de la révision.
- v.
- c. Dans Type de fichier de révision, si un message indique que le type de fichier n'a pas pu être détecté, choisissez .zip.
- d. Laissez Description du déploiement vide.
- e. Étendre les remplacements des groupes de déploiement Dans la liste de configuration du déploiement, sélectionnez CodeDeployDefault. OneAtATime, puis choisissez Create deployment.

Vous pouvez vérifier l'état du déploiement comme décrit dans la rubrique [Surveillez et résolvez les problèmes liés à votre déploiement](#).

Une fois le site CodeDeploy redéployé, revisitez-le dans votre navigateur Web pour vérifier que la couleur d'arrière-plan et le texte de la page Web ont été modifiés. (Vous devrez peut-être actualiser votre navigateur.) Si la couleur d'arrière-plan et le texte ont été modifiés, vous avez réussi la procédure. Vous avez modifié et redéployé votre site.

Étape 6 : Nettoyez votre message « Bonjour tout le monde ! » application et ressources connexes

Vous avez maintenant correctement mis à jour le message « Hello, World ! » codez et redéployez le site. Pour éviter des frais permanents pour les ressources que vous avez créées pour effectuer ce didacticiel, vous devez supprimer :

- Toutes les CloudFormation piles (ou résiliez toute instance Amazon EC2, si vous les avez créées en dehors CloudFormation de).
- Tous les compartiments Amazon S3.
- l'application HelloWorld_App dans CodeDeploy ;
- L'association AWS Systems Manager State Manager de l' CodeDeploy agent.

Vous pouvez utiliser les AWS CLI consoles Amazon S3, Amazon EC2 et les CodeDeploy consoles, ou AWS APIs pour effectuer le nettoyage. CloudFormation

Rubriques

- [Pour nettoyer les ressources \(interface CLI\)](#)
- [Pour nettoyer les ressources \(console\)](#)
- [Quelle est la prochaine étape ?](#)

Pour nettoyer les ressources (interface CLI)

1. Si vous avez utilisé la CloudFormation pile pour ce didacticiel, supprimez-la en appelant la delete-stack commande correspondant à la pile nommée **CodeDeployDemoStack**. Cela met fin à toutes les instances Amazon EC2 associées et supprime tous les rôles IAM associés créés à l'origine par la pile.

```
aws cloudformation delete-stack --stack-name CodeDeployDemoStack
```

2. Pour supprimer le compartiment Amazon S3, appelez la `rm` commande en utilisant le `--recursive` commutateur correspondant au compartiment nommé **amzn-s3-demo-bucket**. Elle supprime le compartiment et tous les objets qui y figurent.

```
aws s3 rm s3://amzn-s3-demo-bucket --recursive --region region
```

3. Pour supprimer l'HelloWorld_App application CodeDeploy, appelez la `delete-application` commande. Elle supprime tous les enregistrements de groupe de déploiement et les enregistrements de déploiement associés pour l'application.

```
aws deploy delete-application --application-name HelloWorld_App
```

4. Pour supprimer l'association Systems Manager State Manager, appelez la `delete-association` commande.

```
aws ssm delete-association --association-id association-id
```

Vous pouvez l'obtenir *association-id* en appelant la `describe-association` commande.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets  
Key=tag:Name,Values=CodeDeployDemo
```

5. Si vous n'avez pas utilisé la CloudFormation pile pour ce didacticiel, appelez la `terminate-instances` commande pour mettre fin aux instances Amazon EC2 que vous avez créées manuellement. Fournissez l'ID de l'instance Amazon EC2 à résilier.

```
aws ec2 terminate-instances --instance-ids instanceId
```

Pour nettoyer les ressources (console)

Si vous avez utilisé notre CloudFormation modèle pour ce didacticiel, supprimez la CloudFormation pile associée.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la zone de recherche, tapez le nom de la CloudFormation pile (par exemple, **CodeDeployDemoStack**).
3. Cochez la case en regard du nom de la pile.

4. Dans le menu Actions, sélectionnez Supprimer pile. Cela supprime la pile, met fin à toutes les instances Amazon EC2 associées et supprime tous les rôles IAM associés.

Pour mettre fin aux instances Amazon EC2 que vous avez créées en dehors d'une CloudFormation pile :

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/ec2/>
2. Dans la zone Instances, choisissez Instances.
3. Dans le champ de recherche, tapez le nom de l'instance Amazon EC2 que vous souhaitez résilier, puis appuyez sur Entrée.
4. Choisissez l'instance Amazon EC2.
5. Choisissez Actions, pointez sur État de l'instance, puis choisissez Terminer. À l'invite, choisissez Oui, résilier. Répétez ces étapes pour toutes les instances Amazon EC2 supplémentaires.

Pour supprimer le compartiment Amazon S3, procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/s3/>.
2. Dans la liste des compartiments, recherchez et choisissez le nom du compartiment Amazon S3 (par exemple, **amzn-s3-demo-bucket**).
3. Avant de pouvoir supprimer un compartiment, vous devez d'abord supprimer son contenu. Sélectionnez tous les fichiers figurant dans le compartiment, tels que **HelloWorld_App.zip**. Dans le menu Actions, sélectionnez Delete. Lorsque vous êtes invité à confirmer la suppression, choisissez OK.
4. Une fois que le compartiment est vide, vous pouvez le supprimer. Dans la liste des compartiments, sélectionnez la ligne du compartiment (mais pas son nom). Choisissez Supprimer le compartiment, lorsque vous êtes invité à confirmer, puis OK.

Pour supprimer l'HelloWorld_Appapplication depuis CodeDeploy :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Sélectionnez **HelloWorld_App**.
4. Choisissez Supprimer l'application.
5. Lorsque vous y êtes invité, tapez **Delete**, puis choisissez Supprimer.

Pour supprimer l'association Systems Manager State Manager, procédez comme suit :

1. Ouvrez la AWS Systems Manager console dans <https://console.aws.amazon.com/systems-manager>.
2. Dans le volet de navigation, sélectionnez State Manager.
3. Sélectionnez l'association que vous avez créée et choisissez Delete (Supprimer).

Quelle est la prochaine étape ?

Si vous êtes arrivé ici, vous avez terminé avec succès un déploiement avec CodeDeploy.
Félicitations !

Tutoriel : Déployer une application sur une instance locale avec CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux)

Ce didacticiel vous aide à acquérir de l'expérience en vous CodeDeploy guidant dans le déploiement d'un exemple de révision d'application sur une seule instance locale, c'est-à-dire un appareil physique autre qu'une instance Amazon EC2, exécutant Windows Server, Ubuntu Server ou Red Hat Enterprise Linux (RHEL). Pour plus d'informations sur les instances locales et leur mode de fonctionnement CodeDeploy, consultez [Working with On-Premises Instances](#).

Ce n'est pas ce que vous recherchez ?

- Pour vous entraîner au déploiement sur une instance Amazon EC2 exécutant Amazon Linux ou RHEL, consultez. [Tutoriel : Déploiement WordPress sur une instance Amazon EC2 \(Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix\)](#)
- Pour vous entraîner au déploiement sur une instance Amazon EC2 exécutant Windows Server, consultez. [Tutoriel : Déployez un message « Bonjour tout le monde ! » application avec CodeDeploy \(Windows Server\)](#)

Rubriques

- [Conditions préalables](#)
- [Étape 1 : Configuration de l'instance sur site](#)
- [Étape 2 : Création d'un exemple de révision d'application](#)
- [Étape 3 : Regroupez et chargez la révision de votre application sur Amazon S3](#)
- [Étape 4 : Déployez la révision de votre application](#)
- [Étape 5 : vérifier votre déploiement](#)
- [Étape 6 : nettoyer les ressources](#)

Conditions préalables

Avant de commencer ce didacticiel, vous devez remplir les conditions requises, notamment la configuration d'un utilisateur [Commencer avec CodeDeploy](#), l'installation ou la mise à niveau du AWS CLI, et la création d'un rôle de service. Il n'est pas nécessaire de créer un profil d'instance IAM comme décrit dans les conditions préalables. Les instances locales n'utilisent pas de profils d'instance IAM.

L'appareil physique que vous allez configurer comme instance sur site doit exécuter l'un des systèmes d'exploitation répertoriés dans [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#).

Étape 1 : Configuration de l'instance sur site

Avant de pouvoir effectuer un déploiement sur votre instance sur site, vous devez la configurer. Suivez les instructions de [Working with On-Premises Instances](#), puis revenez à cette page.

Installation de l' CodeDeploy agent

Après avoir configuré votre instance locale, suivez les étapes relatives aux instances locales dans [Installer l' CodeDeploy agent](#) et revenez à cette page.

Étape 2 : Création d'un exemple de révision d'application

Dans cette étape, vous créez un exemple de révision d'application à déployer sur votre instance sur site.

Comme il est difficile de savoir quels logiciels et quelles fonctionnalités sont déjà installés (ou autorisés à être installés conformément aux politiques de votre organisation) sur votre instance locale, l'exemple de révision d'application que nous proposons ici utilise simplement des scripts batch (pour Windows Server) ou des scripts shell (pour Ubuntu Server et RHEL) pour écrire des fichiers texte sur un emplacement de votre instance locale. Un fichier est écrit pour chacun des événements du cycle de vie du CodeDeploy déploiement, notamment Install AfterInstall, ApplicationStart, et ValidateService. Au cours de l'événement relatif au cycle de vie du BeforeInstalldéploiement, un script sera exécuté pour supprimer les anciens fichiers écrits lors des déploiements précédents de cet exemple et créer un emplacement sur votre instance locale où écrire les nouveaux fichiers.

Note

Cet exemple de révision d'application peut ne pas être déployé si l'une des affirmations suivantes est vraie :

- L'utilisateur qui démarre l' CodeDeploy agent sur l'instance locale n'est pas autorisé à exécuter des scripts.
- L'utilisateur n'est pas autorisé à créer ou à supprimer des dossiers dans les emplacements répertoriés dans les scripts.
- L'utilisateur n'est pas autorisé à créer des fichiers texte dans les emplacements répertoriés dans les scripts.

Note

Si vous avez configuré une instance Windows Server et que vous souhaitez déployer un autre exemple, vous pouvez utiliser celui présenté [Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 de Windows Server](#) dans le [Tutoriel : Déployez](#)

[un message « Bonjour tout le monde ! » application avec CodeDeploy \(Windows Server\)](#) didacticiel.

Si vous avez configuré une instance RHEL et que vous souhaitez déployer un autre exemple, vous pouvez utiliser celui présenté [Étape 2 : configurer votre contenu source à déployer sur l'instance Amazon EC2 Amazon Linux ou Red Hat Enterprise Linux](#) dans le [Tutoriel : Déploiement WordPress sur une instance Amazon EC2 \(Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix\)](#) didacticiel.

Il n'existe actuellement aucun autre exemple pour Ubuntu Server.

1. Sur votre machine de développement, créez un sous-répertoire (sous-dossier) nommé CodeDeployDemo-OnPrem, qui stockera les exemples de fichiers de révision d'application, puis basculez vers ce sous-dossier. Pour cet exemple, nous supposons que vous utiliserez le c : \temp dossier comme dossier racine pour Windows Server ou le /tmp dossier comme dossier racine pour Ubuntu Server et RHEL. Si vous utilisez un autre dossier, veillez à le substituer aux nôtres tout au long de ce didacticiel :

Pour Windows :

```
mkdir c:\temp\CodeDeployDemo-OnPrem
cd c:\temp\CodeDeployDemo-OnPrem
```

Pour Linux, macOS ou Unix :

```
mkdir /tmp/CodeDeployDemo-OnPrem
cd /tmp/CodeDeployDemo-OnPrem
```

2. A la racine du sous-dossier CodeDeployDemo-OnPrem, utilisez un éditeur de texte pour créer deux fichiers nommés appspec.yml et install.txt :

appspec.yml pour Windows Server :

```
version: 0.0
os: windows
files:
  - source: .\install.txt
    destination: c:\temp\CodeDeployExample
hooks:
  BeforeInstall:
```



```
- location: .\scripts\before-install.bat
  timeout: 900
AfterInstall:
- location: .\scripts\after-install.bat
  timeout: 900
ApplicationStart:
- location: .\scripts\application-start.bat
  timeout: 900
ValidateService:
- location: .\scripts\validate-service.bat
  timeout: 900
```

appspec.yml pour Ubuntu Server et RHEL :

```
version: 0.0
os: linux
files:
- source: ./install.txt
  destination: /tmp/CodeDeployExample
hooks:
  BeforeInstall:
    - location: ./scripts/before-install.sh
      timeout: 900
  AfterInstall:
    - location: ./scripts/after-install.sh
      timeout: 900
  ApplicationStart:
    - location: ./scripts/application-start.sh
      timeout: 900
  ValidateService:
    - location: ./scripts/validate-service.sh
      timeout: 900
```

Pour plus d'informations sur AppSpec les fichiers, reportez-vous [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#) aux sections et [CodeDeploy AppSpec référence de fichier](#).

install.txt:

```
The Install deployment lifecycle event successfully completed.
```

3. À la racine du sous-dossier CodeDeployDemo-OnPrem, créez un sous-dossier `scripts`, puis accédez-y :

Pour Windows :

```
mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
cd c:\temp\CodeDeployDemo-OnPrem\scripts
```

Pour Linux, macOS ou Unix :

```
mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
cd /tmp/CodeDeployDemo-OnPrem/scripts
```

4. À la racine du `scripts` sous-dossier, utilisez un éditeur de texte pour créer quatre fichiers nommés `before-install.bat`, `after-install.bat`, `application-start.bat`, et `validate-service.bat` pour Windows Server, ou, `before-install.sh`, `after-install.sh`, `application-start.sh`, et `validate-service.sh` pour Ubuntu Server et RHEL :

Pour Windows Server :

`before-install.bat`:

```
set FOLDER=%HOMEDRIVE%\temp\CodeDeployExample

if exist %FOLDER% (
    rd /s /q "%FOLDER%"
)

mkdir %FOLDER%
```

`after-install.bat`:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The AfterInstall deployment lifecycle event successfully completed. > after-install.txt
```

`application-start.bat`:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ApplicationStart deployment lifecycle event successfully completed. >
application-start.txt
```

validate-service.bat:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ValidateService deployment lifecycle event successfully completed. >
validate-service.txt
```

Pour Ubuntu Server et RHEL :

before-install.sh:

```
#!/bin/bash
export FOLDER=/tmp/CodeDeployExample

if [ -d $FOLDER ]
then
    rm -rf $FOLDER
fi

mkdir -p $FOLDER
```

after-install.sh:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The AfterInstall deployment lifecycle event successfully completed." > after-
install.txt
```

application-start.sh:

```
#!/bin/bash
cd /tmp/CodeDeployExample
```

```
echo "The ApplicationStart deployment lifecycle event successfully completed." >
application-start.txt
```

`validate-service.sh`:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ValidateService deployment lifecycle event successfully completed." >
validate-service.txt

unset FOLDER
```

5. Pour Ubuntu Server et RHEL uniquement, assurez-vous que les quatre scripts shell disposent des autorisations d'exécution :

```
chmod +x ./scripts/*
```

Étape 3 : Regroupez et chargez la révision de votre application sur Amazon S3

Avant de déployer la révision de votre application, vous devez regrouper les fichiers, puis télécharger le bundle de fichiers dans un compartiment Amazon S3. Suivez les instructions de [Créez une application avec CodeDeploy](#) et [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#). (Bien que vous puissiez donner un nom quelconque à l'application et au groupe de déploiement, nous vous recommandons d'utiliser CodeDeploy-OnPrem-App pour le nom de l'application et CodeDeploy-OnPrem-DG pour le nom du groupe de déploiement.) Une fois que vous avez suivi ces instructions, revenez à cette page.

Note

Vous pouvez également télécharger le bundle de fichiers dans un GitHub référentiel et le déployer à partir de là. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy avec GitHub](#).

Étape 4 : Déployez la révision de votre application

Après avoir chargé la révision de votre application dans un compartiment Amazon S3, essayez de la déployer sur votre instance sur site. Suivez les instructions de [Créez un déploiement avec CodeDeploy](#), puis revenez à cette page.

Étape 5 : vérifier votre déploiement

Pour vérifier que le déploiement a réussi, suivez les instructions de [Afficher les détails CodeDeploy du déploiement](#), puis revenez à cette page.

Si le déploiement a réussi, vous trouverez quatre fichiers texte dans le `c:\temp\CodeDeployExample` dossier (pour Windows Server) ou `/tmp/CodeDeployExample` (pour Ubuntu Server et RHEL).

Si le déploiement a échoué, suivez les étapes de dépannage de [View Instance Details](#) et [Résolution des problèmes d'instance](#). Réalisez toutes les corrections requises, regroupez et chargez votre révision d'application, puis réessayez d'effectuer le déploiement.

Étape 6 : nettoyer les ressources

Pour éviter des frais permanents pour les ressources que vous avez créées pour ce didacticiel, supprimez le compartiment Amazon S3 si vous ne l'utilisez plus. Vous pouvez également nettoyer les ressources associées, telles que les enregistrements de l'application et du groupe de déploiement dans CodeDeploy et dans l'instance locale.

Vous pouvez utiliser les consoles Amazon S3 AWS CLI ou une combinaison de celles-ci AWS CLI pour nettoyer les ressources. CodeDeploy

Nettoyer les ressources (CLI)

Pour supprimer le compartiment Amazon S3

- Appelez la commande [rm](#) avec le commutateur `--recursive` sur le compartiment (par exemple, `amzn-s3-demo-bucket`). Le compartiment et tous les objets figurant dans le compartiment sont supprimés.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Pour supprimer les enregistrements de l'application et du groupe de déploiement dans CodeDeploy

- Appelez la commande [delete-application](#) sur l'application (par exemple, CodeDeploy-OnPrem-App). Les enregistrements pour le déploiement et le groupe de déploiement sont supprimés.

```
aws deploy delete-application --application-name your-application-name
```

Pour désenregistrer l'instance locale et supprimer l'utilisateur IAM

- Appelez la commande de [désenregistrement](#) par rapport à l'instance et à la région locales :

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --  
region your-region
```

 Note

Si vous ne souhaitez pas supprimer l'utilisateur IAM associé à cette instance locale, utilisez plutôt l'option `--no-delete-iam-user`.

Pour désinstaller l'agent CodeDeploy et supprimer le fichier de configuration de l'instance locale

- À partir de l'instance locale, appelez la commande de [désinstallation](#) :

```
aws deploy uninstall
```

Vous avez maintenant terminé toutes les étapes de nettoyage des ressources utilisées pour ce didacticiel.

Nettoyer les ressources (console)

Pour supprimer le compartiment Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/s3/>.
2. Choisissez l'icône à côté du compartiment que vous souhaitez supprimer (par exemple, amzn-s3-demo-bucket), mais ne choisissez pas le compartiment lui-même.

3. Choisissez Actions, puis Supprimer.
4. Lorsque vous êtes invité à supprimer le compartiment, choisissez OK.

Pour supprimer les enregistrements de l'application et du groupe de déploiement dans CodeDeploy

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, choisissez Applications.
3. Choisissez le nom de l'application à supprimer (par exemple, CodeDeploy-OnPrem-App), puis choisissez Supprimer l'application.
4. À l'invite, entrez le nom de l'application pour confirmer que vous souhaitez la supprimer, puis choisissez Supprimer.

Vous ne pouvez pas utiliser la AWS CodeDeploy console pour désenregistrer l'instance locale ou désinstaller l'agent. CodeDeploy Suivez les instructions de la section [Pour désenregistrer l'instance locale et supprimer l'utilisateur IAM](#).

Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling

Dans ce didacticiel, vous allez utiliser CodeDeploy pour déployer une révision d'application dans un groupe Auto Scaling. Amazon EC2 Auto Scaling lance des instances Amazon EC2 selon des conditions prédéfinies, puis met fin à ces instances lorsqu'elles ne sont plus nécessaires. Amazon EC2 Auto Scaling peut CodeDeploy vous aider à évoluer en garantissant qu'il dispose toujours du nombre correct d'instances Amazon EC2 disponibles pour gérer la charge des déploiements. Pour plus d'informations sur l'intégration d'Amazon EC2 Auto Scaling CodeDeploy avec, [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#) consultez.

Rubriques

- [Conditions préalables](#)

- [Étape 1 : Création et configuration du groupe Auto Scaling](#)
- [Étape 2 : Déployer l'application dans le groupe Auto Scaling](#)
- [Étape 3 : Vérifiez vos résultats](#)
- [Étape 4 : augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling](#)
- [Étape 5 : Vérifiez à nouveau vos résultats](#)
- [Étape 6 : Nettoyer](#)

Conditions préalables

Pour suivre ce tutoriel :

- Effectuez toutes les étapes décrites [Commencer avec CodeDeploy](#), y compris la configuration AWS CLI et la création d'un profil d'instance IAM (**CodeDeployDemo-EC2-Instance-Profile**) et d'un rôle de service (**CodeDeployDemo**). Un rôle de service est un type spécial de rôle IAM qui autorise un service à agir en votre nom.
- Si vous créez votre groupe Auto Scaling avec un modèle de lancement, vous devez ajouter les autorisations suivantes :
 - `ec2:RunInstances`
 - `ec2:CreateTags`
 - `iam:PassRole`

Pour plus d'informations [Étape 2 : créer un rôle de service](#), consultez les sections [Création d'un modèle de lancement pour un groupe Auto Scaling](#) et [prise en charge des modèles de lancement](#) dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

- Créez et utilisez une révision compatible avec une instance de serveur Ubuntu et CodeDeploy. Pour votre révision, vous pouvez effectuer l'une des opérations suivantes :
 - Créer et utiliser l'exemple de révision de l'[Étape 2 : Création d'un exemple de révision d'application](#) du didacticiel [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#).
 - Créer vous-même une révision. Pour ce faire, consultez [Travailler avec les révisions d'applications pour CodeDeploy](#).
- Créez un groupe de sécurité nommé **CodeDeployDemo-AS-SG** avec la règle entrante suivante :
 - Type : HTTP
 - Source : N'importe où

Cela est nécessaire pour visualiser votre application et vérifier le succès du déploiement. Pour plus d'informations sur la création d'un groupe de sécurité, consultez la section [Création d'un groupe de sécurité](#) dans le guide de l'utilisateur Amazon EC2.

Étape 1 : Création et configuration du groupe Auto Scaling

Au cours de cette étape, vous allez créer un groupe Auto Scaling contenant une seule instance Amazon EC2 Amazon Linux, RHEL ou Windows Server. Dans une étape ultérieure, vous demanderez à Amazon EC2 Auto Scaling d'ajouter une instance Amazon EC2 supplémentaire, CodeDeploy et vous y déployerez votre révision.

Rubriques

- [Pour créer et configurer le groupe Auto Scaling \(CLI\)](#)
- [Pour créer et configurer le groupe Auto Scaling \(console\)](#)

Pour créer et configurer le groupe Auto Scaling (CLI)

1. Appelez la `create-launch-template` commande pour créer un modèle de lancement Amazon EC2.

Avant d'appeler cette commande, vous avez besoin de l'ID d'une AMI compatible avec ce didacticiel, représenté par l'espace réservé `image-id`. Vous avez également besoin du nom d'une paire de clés d'instance Amazon EC2 pour permettre l'accès à l'instance Amazon EC2, représentée par l'espace réservé. `key-name`

Pour obtenir l'ID d'une image AMI qui fonctionne avec ce didacticiel :

- a. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
- b. Dans le volet de navigation, sous Instances, choisissez Instances, puis choisissez Lancer une instance.
- c. Dans l'onglet Démarrage rapide de la page Choisissez une image de machine Amazon, notez l'ID de l'AMI à côté de l'AMI Amazon Linux 2, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS ou Microsoft Windows Server 2012 R2.

 Note

Si vous avez une version personnalisée d'une image AMI compatible avec CodeDeploy choisissez-la ici au lieu de parcourir l'onglet Démarrage rapide. Pour plus d'informations sur l'utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling, [Utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling](#) consultez.

Pour la paire de clés d'instance Amazon EC2, utilisez le nom de votre paire de clés d'instance Amazon EC2.

Appelez la commande `create-launch-template`.

Sur les machines Linux, macOS ou Unix locales :

```
aws ec2 create-launch-template \  
  --launch-template-name CodeDeployDemo-AS-Launch-Template \  
  --launch-template-data file://config.json
```

Le contenu du `config.json` fichier :

```
{  
  "InstanceType": "t1.micro",  
  "ImageId": "image-id",  
  "IamInstanceProfile": {  
    "Name": "CodeDeployDemo-EC2-Instance-Profile"  
  },  
  "KeyName": "key-name"  
}
```

Sur les machines Windows locales :

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-  
Template --launch-template-data file://config.json
```

Le contenu du `config.json` fichier :

```
{
  "InstanceType": "t1.micro",
  "ImageId": "image-id",
  "IamInstanceProfile": {
    "Name": "CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName": "key-name"
}
```

Ces commandes, associées au `config.json` fichier, créent un modèle de lancement Amazon EC2 nommé `CodeDeployDemo-AS-Launch-Template` d'après votre groupe Auto Scaling qui sera créé lors d'une étape suivante sur la base du type d'instance Amazon EC2 `t1.micro`. Sur la base de vos entrées pour `ImageId`, `IamInstanceProfile`, et `KeyName`, le modèle de lancement spécifie également l'ID AMI, le nom du profil d'instance associé au rôle IAM à transmettre aux instances lors du lancement et la paire de clés Amazon EC2 à utiliser lors de la connexion aux instances.

2. Appelez la `create-auto-scaling-group` commande pour créer un groupe Auto Scaling. Vous aurez besoin du nom de l'une des zones de disponibilité dans l'une des régions répertoriées dans [Région et des points de terminaison](#) dans la Références générales AWS, représentés par l'espace réservé *availability-zone*.

Note

Pour afficher la liste des zones de disponibilité d'une région, appelez :

```
aws ec2 describe-availability-zones --region region-name
```

Par exemple, pour consulter la liste des zones de disponibilité dans la région ouest des États-Unis (Oregon), appelez :

```
aws ec2 describe-availability-zones --region us-west-2
```

Pour obtenir la liste des identifiants de noms de régions, consultez [Noms des compartiments du kit de ressources par région](#).

Sur les machines Linux, macOS ou Unix locales :

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name CodeDeployDemo-AS-Group \  
  --launch-template CodeDeployDemo-AS-Launch-Template,Version='$Latest' \  
  --min-size 1 \  
  --max-size 1 \  
  --desired-capacity 1 \  
  --availability-zones availability-zone \  
  --tags Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Sur les machines Windows locales :

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name  
CodeDeployDemo-AS-Group --launch-template LaunchTemplateName=CodeDeployDemo-  
AS-Launch-Template,Version="$Latest" --min-size 1 --max-size 1 --  
desired-capacity 1 --availability-zones availability-zone --tags  
Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Ces commandes créent un groupe Auto Scaling nommé d'**CodeDeployDemo-AS-Group** après le modèle de lancement Amazon EC2 nommé. **CodeDeployDemo-AS-Launch-Template**. Ce groupe Auto Scaling ne possède qu'une seule instance Amazon EC2 et il est créé dans la zone de disponibilité spécifiée. Chaque instance de ce groupe Auto Scaling aura le `tagName=CodeDeployDemo`. La balise sera utilisée lors de l'installation ultérieure de l'CodeDeploy agent.

3. Appelez la commande `describe-auto-scaling-groups` sur **CodeDeployDemo-AS-Group** :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names  
CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,  
LifecycleState]" --output text
```

Ne continuez pas tant que les valeurs retournées n'affichent pas `Healthy` et `InService`.

4. L'CodeDeploy agent doit être installé sur les instances de votre groupe Auto Scaling pour être utilisé dans CodeDeploy les déploiements. Installez l'CodeDeploy agent en appelant la `create-association` commande depuis AWS Systems Manager les balises ajoutées lors de la création du groupe Auto Scaling.

```
aws ssm create-association \  
  --name AWS-ConfigureAWSPackage \  
  --targets Key=tag:Name,Values=CodeDeployDemo \  

```

```
--parameters action=Install, name=AWSCodeDeployAgent \  
--schedule-expression "cron(0 2 ? * SUN *)"
```

Cette commande crée une association dans Systems Manager State Manager qui installera l'CodeDeploy agent sur toutes les instances du groupe Auto Scaling, puis tentera de le mettre à jour à 2 h tous les dimanches matins. Pour plus d'informations sur l'CodeDeploy agent, consultez la section [Utilisation de l'CodeDeploy agent](#). Pour plus d'informations sur Systems Manager, consultez [What is AWS Systems Manager](#).

Pour créer et configurer le groupe Auto Scaling (console)

1. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
2. Dans la barre de navigation globale, assurez-vous que l'une des régions répertoriées dans [Région et points de terminaison](#) dans le Références générales AWS est sélectionnée. Les ressources Amazon EC2 Auto Scaling sont liées à la région que vous spécifiez CodeDeploy et ne sont prises en charge que dans certaines régions.
3. Dans la barre de navigation, sous Instances, choisissez Launch Templates.
4. Choisissez Créer un modèle de lancement.
5. Dans la boîte de dialogue Nom et description du modèle de lancement, pour Nom du modèle de lancement, entrez **CodeDeployDemo-AS-Launch-Template**. Conservez les valeurs par défaut pour les autres champs.
6. Dans la boîte de dialogue Amazon machine image (AMI), cliquez sur le menu déroulant sous AMI, choisissez une AMI compatible avec ce didacticiel :
 - Dans l'onglet Quick Start de la liste déroulante AMI, sélectionnez l'une des options suivantes : Amazon Linux 2 AMI, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS ou Microsoft Windows Server 2012 R2.

Note

Si vous avez une version personnalisée d'une image AMI compatible avec CodeDeploy choisissez-la ici au lieu de parcourir l'onglet Démarrage rapide. Pour plus d'informations sur l'utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling, [Utilisation d'une AMI personnalisée avec CodeDeploy Amazon EC2 Auto Scaling](#) consultez.

7. Dans Type d'instance, sélectionnez le menu déroulant et choisissez t1.micro. Vous pouvez utiliser la barre de recherche pour le trouver plus rapidement.
8. Dans la boîte de dialogue Key pair (login), sélectionnez Choisir une paire de clés existante. Dans la liste déroulante Sélectionnez une paire de clés, sélectionnez la paire de clés d'instance Amazon EC2 que vous avez créée ou utilisée lors des étapes précédentes.
9. Dans la boîte de dialogue Paramètres réseau, choisissez Virtual Public Cloud (VPC).

Dans le menu déroulant Groupes de sécurité, choisissez le groupe de sécurité que vous avez créé dans la [section des prérequis du didacticiel](#) (**CodeDeployDemo-AS-SG**).

10. Développez la boîte de dialogue Détails avancés. Dans la liste déroulante du profil d'instance IAM, sélectionnez le rôle IAM que vous avez créé précédemment (**CodeDeployDemo-EC2-Instance-Profile**) sous Profil d'instance IAM.

Conservez le reste des valeurs par défaut.

11. Choisissez Créer un modèle de lancement.
12. Dans la boîte de dialogue Next steps, choisissez Create Auto Scaling group.
13. Sur la page Choisir un modèle de lancement ou de configuration, saisissez le nom du groupe Auto Scaling **CodeDeployDemo-AS-Group**.
14. Dans la boîte de dialogue Modèle de lancement, votre modèle de lancement (**CodeDeployDemo-AS-Launch-Template**) doit être renseigné. Sinon, sélectionnez-le dans le menu déroulant. Conservez les valeurs par défaut et choisissez Next.
15. Sur la page Choisir les options de lancement de l'instance, dans la section Réseau, pour VPC, choisissez le VPC par défaut. Ensuite, pour les zones de disponibilité et les sous-réseaux, choisissez un sous-réseau par défaut. Vous devez créer un VPC si vous ne pouvez pas choisir la valeur par défaut. Pour plus d'informations, consultez [Getting started with Amazon VPC](#).
16. Dans la section Instance type requirements (Exigences relatives au type d'instance), utilisez le paramètre par défaut pour simplifier cette étape. (Ne remplacez pas le modèle de lancement.) Pour ce didacticiel, vous lancerez uniquement des instances à la demande en utilisant le type d'instance spécifié dans votre modèle de lancement.
17. Choisissez Next (Suivant) pour accéder à la page Configurer les options avancées.
18. Conservez les valeurs par défaut et choisissez Next.
19. Sur la page Configurer la taille du groupe et les politiques de dimensionnement, conservez les valeurs de taille de groupe par défaut de 1. Choisissez Suivant.
20. Ignorez l'étape de configuration des notifications et choisissez Next.

21. Sur la page Ajouter des balises, ajoutez une balise à utiliser ultérieurement lors de l'installation de l' CodeDeploy agent. Choisissez Ajouter une balise.
 - a. Dans Clé, entrez **Name**.
 - b. Dans Value (Valeur), entrez **CodeDeployDemo**.
- Choisissez Suivant.
22. Passez en revue les informations de votre groupe Auto Scaling sur la page de révision, puis choisissez Create Auto Scaling group.
23. Dans la barre de navigation, avec Auto Scaling Groups sélectionné **CodeDeployDemo-AS-Group**, choisissez, puis choisissez l'onglet Instance Management. Ne poursuivez pas tant que la valeur de n'InService apparaît pas dans la colonne Lifecycle et que la valeur de Healthy n'apparaît pas dans la colonne Health Status.
24. Installez l' CodeDeploy agent en suivant les étapes décrites dans [Installer l' CodeDeploy agent](#) et en utilisant les balises d'Name=CodeDeployDemoinstance.

Étape 2 : Déployer l'application dans le groupe Auto Scaling

Au cours de cette étape, vous allez déployer la révision sur l'instance Amazon EC2 unique du groupe Auto Scaling.

Rubriques

- [Pour créer le déploiement \(interface CLI\)](#)
- [Pour créer le déploiement \(console\)](#)

Pour créer le déploiement (interface CLI)

1. Appelez la commande create-application pour créer une application nommée **SimpleDemoApp** :

```
aws deploy create-application --application-name SimpleDemoApp
```

2. Vous devriez avoir déjà créé un rôle de service en suivant les instructions de la rubrique [Étape 2 : créer un rôle de service pour CodeDeploy](#). Le rôle de service donnera CodeDeploy l'autorisation d'accéder à vos instances Amazon EC2 pour développer (lire) leurs balises. Vous avez besoin du nom ARN du rôle de service. Afin d'obtenir le nom ARN du rôle de service, suivez les instructions de la rubrique [Obtenez l'ARN du rôle de service \(CLI\)](#).

- Maintenant que vous disposez d'un ARN de rôle de service, appelez la `create-deployment-group` commande pour créer un groupe de déploiement nommé **SimpleDemoDG**, associé à l'application nommée **SimpleDemoApp**, en utilisant le nom du groupe Auto Scaling **CodeDeployDemo-AS-Group** et le nom de la configuration de déploiement **CodeDeployDefault.OneAtATime**, avec l'ARN du rôle de service spécifié.

 Note

La [create-deployment-group](#) commande permet de créer des déclencheurs qui entraînent l'envoi de notifications Amazon SNS aux abonnés du topic concernant des événements spécifiques liés aux déploiements et aux instances. La commande prend également en charge des options permettant d'annuler automatiquement les déploiements et de configurer des alarmes pour arrêter les déploiements lorsque les seuils de surveillance des CloudWatch alarmes d'Amazon sont atteints. Les commandes associées à ces actions ne sont pas incluses dans ce didacticiel.

Sur les machines Linux, macOS ou Unix locales :

```
aws deploy create-deployment-group \  
  --application-name SimpleDemoApp \  
  --auto-scaling-groups CodeDeployDemo-AS-Group \  
  --deployment-group-name SimpleDemoDG \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --service-role-arn service-role-arn
```

Sur les machines Windows locales :

```
aws deploy create-deployment-group --application-name SimpleDemoApp --auto-scaling-  
groups CodeDeployDemo-AS-Group --deployment-group-name SimpleDemoDG --deployment-  
config-name CodeDeployDefault.OneAtATime --service-role-arn service-role-arn
```

- Appellez la commande `create-deployment` pour créer un déploiement associé à l'application nommée **SimpleDemoApp**, à la configuration de déploiement nommée **CodeDeployDefault.OneAtATime**, au groupe de déploiement nommé **SimpleDemoDG**, en utilisant la révision à l'emplacement spécifié.

Pour les instances Amazon Linux et RHEL Amazon EC2, appels depuis des machines Linux, macOS ou Unix locales


```
aws deploy create-deployment \  
  --application-name SimpleDemoApp \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name SimpleDemoDG \  
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/  
SampleApp_Linux.zip
```

bucket-name est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio), remplacez *bucket-name* par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

Pour les instances Amazon Linux et RHEL Amazon EC2, appels depuis des machines Windows locales

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-  
name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-  
location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Linux.zip
```

bucket-name est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio), remplacez *bucket-name* par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

Pour les instances Amazon EC2 de Windows Server, appels depuis des machines Linux, macOS ou Unix locales

```
aws deploy create-deployment \  
  --application-name SimpleDemoApp \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name SimpleDemoDG \  
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/  
SampleApp_Windows.zip
```

bucket-name est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio),

remplacez *bucket-name* par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

Pour les instances Amazon EC2 de Windows Server, appel depuis des machines Windows locales

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Windows.zip
```

bucket-name est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio), remplacez *bucket-name* par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

 Note

Actuellement, CodeDeploy ne fournit pas d'exemple de révision à déployer sur les instances Amazon EC2 du serveur Ubuntu. Pour créer une révision par vous-même, consultez [Travailler avec les révisions d'applications pour CodeDeploy](#).

5. Appelez la commande `get-deployment` pour vous assurer que le déploiement a réussi.

Avant d'appeler cette commande, vous avez besoin de l'ID du déploiement, qui doit avoir été renvoyé par l'appel à la commande `create-deployment`. Si vous avez besoin d'obtenir l'ID de déploiement à nouveau, appelez la commande `list-deployments` sur l'application nommée **SimpleDemoApp** et le groupe de déploiement nommé **SimpleDemoDG** :

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-name SimpleDemoDG --query "deployments" --output text
```

Maintenant, appelez la commande `get-deployment`, en utilisant l'ID du déploiement :

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.status" --output text
```

Ne continuez pas tant que la valeur `Succeeded` n'est pas retournée.

Pour créer le déploiement (console)

1. Vous devriez avoir déjà créé un rôle de service en suivant les instructions de la rubrique [Étape 2 : créer un rôle de service pour CodeDeploy](#). Le rôle de service donnera CodeDeploy l'autorisation d'accéder à vos instances pour développer (lire) leurs balises. Avant d'utiliser la CodeDeploy console pour déployer la révision de votre application, vous aurez besoin de l'ARN du rôle de service. Afin d'obtenir le nom ARN du rôle de service, suivez les instructions de la rubrique [Obtenir l'ARN du rôle de service \(console\)](#).
2. Maintenant que vous disposez de l'ARN du rôle de service, vous pouvez utiliser la CodeDeploy console pour déployer la révision de votre application.

Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

3. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
4. Choisissez Créer une application.
5. Choisissez Application personnalisée.
6. Dans Nom de l'application, saisissez **SimpleDemoApp**.
7. Dans Plateforme de calcul, choisissez EC2/Solutions sur site.
8. Choisissez Créer une application.
9. Dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement.
10. Dans Nom du groupe de déploiement, saisissez **SimpleDemoDG**.
11. Dans Rôle de service, choisissez le nom du rôle de service .
12. Dans Type de déploiement, choisissez Sur place.
13. Dans Configuration de l'environnement, sélectionnez les groupes Auto Scaling, puis choisissez **CodeDeployDemo-AS-Group**.
14. Dans Configuration du déploiement, choisissez CodeDeployDefault. OneAtATime.
15. Désactivez Activer l'équilibrage de charge.
16. Choisissez Créer un groupe de déploiement.

17. Sur la page du groupe de déploiement, choisissez Créer un déploiement.
18. Dans Type de révision, choisissez Mon application est stockée dans Amazon S3.
19. Dans Emplacement de la révision, entrez l'emplacement de l'exemple d'application pour votre système d'exploitation et votre région.

Pour les instances Amazon Linux et RHEL Amazon EC2

Région	Emplacement de l'exemple d'application
Région US East (Ohio)	<code>http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip</code>
Région US East (N. Virginia)	<code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip</code>
Région US West (N. California)	<code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip</code>
Région US West (Oregon)	<code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip</code>
Région Canada (Centre)	<code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip</code>
Région Europe (Irlande)	<code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip</code>

Région	Emplacement de l'exemple d'application
Région Europe (Londres)	<code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip</code>
Région Europe (Paris)	<code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip</code>
Région Europe (Francfort)	<code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip</code>
Région Israël (Tel Aviv)	<code>https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip</code>
Région Asie-Pacifique (Hong Kong)	<code>https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Linux.zip</code>
Asia Pacific (Tokyo) Region	<code>http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip</code>
Asia Pacific (Seoul) Region	<code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip</code>

Région	Emplacement de l'exemple d'application
Région Asie-Pacifique (Singapour)	<code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip</code>
Région Asia Pacific (Sydney)	<code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip</code>
Région Asie-Pacifique (Melbourne)	<code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip</code>
Région Asie-Pacifique (Mumbai)	<code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip</code>
Région Amérique du Sud (São Paulo)	<code>http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip</code>

Pour les instances Amazon EC2 de Windows Server

Région	Emplacement de l'exemple d'application
Région US East (Ohio)	<code>http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip</code>

Région	Emplacement de l'exemple d'application
Région US East (N. Virginia)	<code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip</code>
Région US West (N. California)	<code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip</code>
Région US West (Oregon)	<code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip</code>
Région Canada (Centre)	<code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip</code>
Région Europe (Irlande)	<code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip</code>
Région Europe (Londres)	<code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip</code>
Région Europe (Paris)	<code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip</code>

Région	Emplacement de l'exemple d'application
Région Europe (Francfort)	<code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip</code>
Région Israël (Tel Aviv)	<code>https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>
Région Asie-Pacifique (Hong Kong)	<code>https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>
Région Asie-Pacifique (Séoul)	<code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip</code>
Région Asie-Pacifique (Singapour)	<code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip</code>
Région Asia Pacific (Sydney)	<code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip</code>
Région Asie-Pacifique (Melbourne)	<code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>

Région	Emplacement de l'exemple d'application
Région Asie-Pacifique (Mumbai)	<code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip</code>
Région Amérique du Sud (São Paulo)	<code>http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip</code>

Pour les instances Amazon EC2 du serveur Ubuntu

Entrez l'emplacement de la révision personnalisée de votre application stockée dans Amazon S3.

20. Laissez Description du déploiement vide.
21. Développez Avancé.
22. Choisissez Créer un déploiement.

 Note

Si Échec s'affiche au lieu de Réussi, vous voudrez peut-être essayer certaines techniques de la rubrique [Surveillez et résolvez les problèmes liés à votre déploiement](#) (en utilisant le nom d'application **SimpleDemoApp** et le nom de groupe de déploiement **SimpleDemoDG**).

Étape 3 : Vérifiez vos résultats

Au cours de cette étape, vous allez vérifier que la **SimpleDemoApp** révision a CodeDeploy été installée sur l'unique instance Amazon EC2 du groupe Auto Scaling.

Rubriques

- [Pour vérifier les résultats \(interface CLI\)](#)
- [Pour vérifier les résultats \(console\)](#)

Pour vérifier les résultats (interface CLI)

Tout d'abord, vous aurez besoin du DNS public de l'instance Amazon EC2.

Utilisez le AWS CLI pour obtenir le DNS public de l'instance Amazon EC2 dans le groupe Auto Scaling en appelant la `describe-instances` commande.

Avant d'appeler cette commande, vous aurez besoin de l'ID de l'instance Amazon EC2. Pour obtenir l'ID, appelez la `describe-auto-scaling-groups` sur **CodeDeployDemo-AS-Group** comme précédemment :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Maintenant, appelez la commande `describe-instances` :

```
aws ec2 describe-instances --instance-id instance-id --query "Reservations[0].Instances[0].PublicDnsName" --output text
```

La valeur renvoyée est le DNS public de l'instance Amazon EC2.

À l'aide d'un navigateur Web, affichez la SimpleDemoApp révision déployée sur cette instance Amazon EC2, à l'aide d'une URL comme celle-ci :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si vous voyez la page de félicitations, cela signifie que vous avez réussi CodeDeploy à déployer une révision sur une seule instance Amazon EC2 dans un groupe Auto Scaling !

Vous allez ensuite ajouter une instance Amazon EC2 au groupe Auto Scaling. Une fois qu'Amazon EC2 Auto Scaling aura ajouté l'instance Amazon EC2, votre révision sera déployée sur la nouvelle instance.

Pour vérifier les résultats (console)

Tout d'abord, vous aurez besoin du DNS public de l'instance Amazon EC2.

Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.

Dans le volet de navigation Amazon EC2, sous Auto Scaling, choisissez Auto Scaling Groups, puis choisissez l'**CodeDeployDemo-AS-Group** entrée.

Dans l'onglet Instances, choisissez l'ID d'instance Amazon EC2 dans la liste.

Sur la page Instances, dans l'onglet Description, notez la valeur DNS public. Elle doit se présenter comme suit : **ec2-01-234-567-890.compute-1.amazonaws.com**

À l'aide d'un navigateur Web, affichez la SimpleDemoApp révision déployée sur cette instance Amazon EC2, à l'aide d'une URL comme celle-ci :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si vous voyez la page de félicitations, cela signifie que vous avez réussi CodeDeploy à déployer une révision sur une seule instance Amazon EC2 dans un groupe Auto Scaling !

Ensuite, vous ajoutez une instance Amazon EC2 au groupe Auto Scaling. Une fois qu'Amazon EC2 Auto Scaling aura ajouté l'instance CodeDeploy Amazon EC2, votre révision sera déployée sur la nouvelle instance Amazon EC2.

Étape 4 : augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling

Au cours de cette étape, vous demandez au groupe Auto Scaling de créer une instance Amazon EC2 supplémentaire. Une fois qu'Amazon EC2 Auto Scaling a créé l'instance CodeDeploy, y déploie votre révision.

Rubriques

- [Pour augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling \(CLI\)](#)
- [Pour augmenter le nombre d'instances Amazon EC2 dans le groupe de déploiement \(console\)](#)

Pour augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling (CLI)

1. Appelez la `update-auto-scaling-group` commande pour augmenter le nombre d'instances Amazon EC2 dans le groupe Auto Scaling nommé **CodeDeployDemo-AS-Group** de un à deux.

Sur les machines Linux, macOS ou Unix locales :

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name CodeDeployDemo-AS-Group \  
  --min-size 2 \  

```

```
--max-size 2 \  
--desired-capacity 2
```

Sur les machines Windows locales :

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-  
AS-Group --min-size 2 --max-size 2 --desired-capacity 2
```

2. Assurez-vous que le groupe Auto Scaling dispose désormais de deux instances Amazon EC2. Appelez la commande `describe-auto-scaling-groups` sur **CodeDeployDemo-AS-Group** :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names  
CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,  
LifecycleState]" --output text
```

Ne continuez pas tant que les deux valeurs retournées n'affichent pas `Healthy` et `InService`.

Pour augmenter le nombre d'instances Amazon EC2 dans le groupe de déploiement (console)

1. Dans la barre de navigation d'Amazon EC2, sous Auto Scaling, choisissez Auto Scaling Groups, puis choisissez **CodeDeployDemo-AS-Group**.
2. Sélectionnez Actions, puis Edit (Modifier).
3. Dans l'onglet Détails, dans les zones Souhaitée, Min et Max, tapez **2**, puis choisissez Enregistrer.
4. Choisissez l'onglet Instances. La nouvelle instance Amazon EC2 doit apparaître dans la liste. (Si l'instance n'apparaît pas, vous pouvez être amené à choisir le bouton Actualiser plusieurs fois.) Ne poursuivez pas tant que la valeur de `n'InService` apparaît pas dans la colonne Lifecycle et que la valeur de `Healthy` n'apparaît pas dans la colonne Health Status.

Étape 5 : Vérifiez à nouveau vos résultats

Au cours de cette étape, vous allez vérifier si la SimpleDemoApp révision a CodeDeploy été installée sur la nouvelle instance du groupe Auto Scaling.

Rubriques

- [Pour vérifier les résultats de déploiement automatique \(interface CLI\)](#)

- [Pour vérifier les résultats de déploiement automatique \(console\)](#)

Pour vérifier les résultats de déploiement automatique (interface CLI)

1. Avant d'appeler la commande `get-deployment`, vous avez besoin de l'ID du déploiement automatique. Pour obtenir l'ID, appelez la commande `list-deployments` sur le l'application nommée **SimpleDemoApp** et le groupe de déploiement nommé **SimpleDemoDG** :

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-name SimpleDemoDG --query "deployments" --output text
```

Il devrait y avoir deux déploiements IDs. Utilisez celui que vous n'avez pas encore utilisé dans un appel à la commande `get-deployment` :

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.[status, creator]" --output text
```

Outre l'état du déploiement, vous devriez le voir `autoScaling` dans le résultat de la commande. (`autoScaling` signifie qu'Amazon EC2 Auto Scaling a créé le déploiement.)

Ne continuez pas tant que l'état du déploiement n'affiche pas `Succeeded`.

2. Avant d'appeler la `describe-instances` commande, vous aurez besoin de l'ID de la nouvelle instance Amazon EC2. Pour obtenir cet ID, effectuez un autre appel de la commande `describe-auto-scaling-groups` sur **CodeDeployDemo-AS-Group** :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Maintenant, appelez la commande `describe-instances` :

```
aws ec2 describe-instances --instance-id instance-id --query "Reservations[0].Instances[0].PublicDnsName" --output text
```

Dans le résultat de la `describe-instances` commande, notez le DNS public de la nouvelle instance Amazon EC2.

3. À l'aide d'un navigateur Web, affichez la SimpleDemoApp révision déployée sur cette instance Amazon EC2, à l'aide d'une URL comme celle-ci :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si la page de félicitations apparaît, cela signifie que vous avez CodeDeploy l'habitude de déployer une révision d'une instance Amazon EC2 agrandie dans un groupe Auto Scaling !

Pour vérifier les résultats de déploiement automatique (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Deployments.
3. Choisissez l'ID de déploiement du déploiement créé par Amazon EC2 Auto Scaling.
.
4. La page Déploiement affiche les informations sur le déploiement. Normalement, vous devez créer un déploiement vous-même, mais Amazon EC2 Auto Scaling en a créé un en votre nom pour déployer votre révision sur la nouvelle instance Amazon EC2.
5. Une fois que la mention Réussi est affichée en haut de la page, vérifiez les résultats sur l'instance. Tout d'abord, vous devez obtenir le DNS public de l'instance :
6. Dans le volet de navigation Amazon EC2, sous Auto Scaling, choisissez Auto Scaling Groups, puis choisissez l'**CodeDeployDemo-AS-Group**entrée.
7. Dans l'onglet Instances, choisissez l'ID de la nouvelle instance Amazon EC2.
8. Sur la page Instances, dans l'onglet Description, notez la valeur DNS public. Elle doit se présenter comme suit : **ec2-01-234-567-890.compute-1.amazonaws.com**

Affichez la révision SimpleDemoApp déployée sur l'instance à l'aide d'une URL similaire à la suivante :

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si la page de félicitations apparaît, cela signifie que vous avez CodeDeploy l'habitude de déployer une révision d'une instance Amazon EC2 agrandie dans un groupe Auto Scaling !

Étape 6 : Nettoyer

Au cours de cette étape, vous allez supprimer le groupe Auto Scaling afin d'éviter des frais permanents pour les ressources que vous avez utilisées au cours de ce didacticiel. Vous pouvez éventuellement supprimer les enregistrements des composants de configuration et de CodeDeploy déploiement d'Auto Scaling.

Rubriques

- [Pour nettoyer les ressources \(interface CLI\)](#)
- [Pour nettoyer les ressources \(console\)](#)

Pour nettoyer les ressources (interface CLI)

1. Supprimez le groupe Auto Scaling en appelant la `delete-auto-scaling-group` commande against **CodeDeployDemo-AS-Group**. Cela va également résilier les instances Amazon EC2.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --force-delete
```

2. Supprimez éventuellement le modèle de lancement Auto Scaling en appelant la `delete-launch-template` commande correspondant à la configuration de lancement nommée **CodeDeployDemo-AS-Launch-Template** :

```
aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template
```

3. Vous pouvez éventuellement supprimer l'application CodeDeploy en appelant la `delete-application` commande correspondant à l'application nommée **SimpleDemoApp**. Cela supprime également tous les enregistrements de révision, de groupe de déploiement et de déploiement associés.

```
aws deploy delete-application --application-name SimpleDemoApp
```

4. Pour supprimer l'association Systems Manager State Manager, appelez la delete-association commande.

```
aws ssm delete-association --association-id association-id
```

Vous pouvez l'obtenir *association-id* en appelant la describe-association commande.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets  
Key=tag:Name,Values=CodeDeployDemo
```

Pour nettoyer les ressources (console)

Pour supprimer le groupe Auto Scaling, qui met également fin aux instances Amazon EC2 :

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse.
<https://console.aws.amazon.com/ec2/>
2. Dans le volet de navigation Amazon EC2, sous Auto Scaling, choisissez Auto Scaling Groups, puis choisissez l'**CodeDeployDemo-AS-Group** entrée.
3. Choisissez Actions, Supprimer, puis Oui, supprimer.

(Facultatif) Pour supprimer le modèle de lancement :

1. Dans la barre de navigation, sous Auto Scaling, choisissez Launch Configurations, puis choisissez **CodeDeployDemo-AS-Launch-Template**.
 2. Choisissez Actions, Supprimer la configuration de lancement, puis Oui, supprimer.
-
1. Vous pouvez éventuellement supprimer l'application de CodeDeploy. Cela supprime également tous les enregistrements de révision, de groupe de déploiement et de déploiement associés. Ouvrez la CodeDeploy console à l'adresse <https://console.aws.amazon.com/codedeploy>.
 2. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

Dans le volet de navigation, développez Deploy, puis choisissez Applications.

3. Dans la liste des applications, choisissez SimpleDemoApp.
4. Sur la page Détails de l'application, choisissez Supprimer l'application.
5. Lorsque vous y êtes invité, tapez **Delete**, puis choisissez Supprimer.

Pour supprimer l'association Systems Manager State Manager, procédez comme suit :

1. Ouvrez la AWS Systems Manager console dans <https://console.aws.amazon.com/systems-manager>.
2. Dans le volet de navigation, sélectionnez State Manager.
3. Sélectionnez l'association que vous avez créée et choisissez Delete (Supprimer).

Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub

Dans ce didacticiel, vous pouvez CodeDeploy déployer un exemple de révision GitHub d'application depuis une seule instance Amazon EC2 exécutant Amazon Linux, une seule instance Red Hat Enterprise Linux (RHEL) ou une seule instance Windows Server. Pour plus d'informations sur GitHub l'intégration avec CodeDeploy, consultez [Intégration CodeDeploy avec GitHub](#).

 Note

Vous pouvez également l'utiliser CodeDeploy pour déployer une révision d'application depuis GitHub une instance de serveur Ubuntu. Vous pouvez utiliser l'exemple de révision décrit [Étape 2 : Création d'un exemple de révision d'application](#) dans [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#), ou vous pouvez créer une révision compatible avec une instance de serveur Ubuntu et CodeDeploy. Pour créer votre propre révision, consultez les pages

[Planifier une révision pour CodeDeploy](#) et [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

Rubriques

- [Conditions préalables](#)
- [Étape 1 : créer un GitHub compte](#)
- [Étape 2 : Création d'un GitHub référentiel](#)
- [Étape 3 : télécharger un exemple d'application dans votre GitHub référentiel](#)
- [Étape 4 : provisionner une instance](#)
- [Étape 5 : Création d'une application et d'un groupe de déploiement](#)
- [Étape 6 : Déployer l'application sur l'instance](#)
- [Étape 7 : Surveiller et vérifier le déploiement](#)
- [Étape 8 : nettoyer](#)

Conditions préalables

Avant de commencer ce didacticiel, procédez comme suit :

- Installez Git sur votre machine locale. Pour installer Git, consultez la section [Téléchargements de Git](#).
- Suivez les étapes de la rubrique [Commencer avec CodeDeploy](#), y compris l'installation et la configuration de l' AWS CLI. Cela est particulièrement important si vous souhaitez utiliser le AWS CLI pour déployer une révision depuis GitHub l'instance.

Étape 1 : créer un GitHub compte

Vous aurez besoin d'un GitHub compte pour créer un GitHub dépôt dans lequel la révision sera stockée. Si vous avez déjà un GitHub compte, passez directement à [Étape 2 : Création d'un GitHub référentiel](#).

1. Accédez à <https://github.com/join>.
2. Tapez un nom d'utilisateur, votre adresse e-mail et un mot de passe.

3. Choisissez S'inscrire pour GitHub, puis suivez les instructions.

Étape 2 : Création d'un GitHub référentiel

Vous aurez besoin d'un GitHub dépôt pour stocker la révision.

Si vous possédez déjà un GitHub dépôt, veillez à le remplacer par son nom

CodeDeployGitHubDemo tout au long de ce didacticiel, puis passez à [Étape 3 : télécharger un exemple d'application dans votre GitHub référentiel](#).

1. Sur la [page d'GitHub accueil](#), effectuez l'une des opérations suivantes :
 - Dans Your repositories, choisissez New repository.
 - Dans la barre de navigation, choisissez Create new (+), puis New repository.
2. Sur la page Create a new repository, procédez comme suit :
 - Dans la zone de texte Nom du référentiel, entrez **CodeDeployGitHubDemo**.
 - Sélectionnez Public.

Note

La sélection de l'option par défaut Public signifie que tout le monde peut voir ce référentiel. Vous pouvez sélectionner l'option Private pour limiter les personnes autorisées à voir et à valider le référentiel.

- Décochez la case Initialize this repository with a README. Vous allez créer un fichier README.md manuellement à la place, à l'étape suivante.
 - Choisissez Créer un référentiel.
3. Suivez les instructions pour que votre type d'ordinateur local puisse utiliser la ligne de commande pour créer le référentiel.

Note

Si vous avez activé l'authentification à deux facteurs GitHub, assurez-vous de saisir votre jeton d'accès personnel au lieu de votre mot de passe de GitHub connexion si vous y êtes invité. Pour plus d'informations, consultez la section [Fourniture de votre code d'authentification 2FA](#).

Sur les machines Linux, macOS ou Unix locales :

1. Depuis le terminal, exécutez les commandes suivantes, une par une, où se *user-name* trouve votre nom GitHub d'utilisateur :

```
mkdir /tmp/CodeDeployGitHubDemo
```

```
cd /tmp/CodeDeployGitHubDemo
```

```
touch README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```

```
git push -u origin master
```

2. Laissez le terminal ouvert à l'emplacement /tmp/CodeDeployGitHubDemo.

Sur les machines Windows locales :

1. A partir d'une invite de commandes exécutée avec des autorisations d'administrateur, exécutez les commandes suivantes, l'une après l'autre :

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. Dans le Bloc-notes, enregistrez le fichier README.md. Fermez le Bloc-notes. Exécutez les commandes suivantes, une par une, en indiquant *user-name* votre nom GitHub d'utilisateur :

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```

```
git push -u origin master
```

3. Laissez l'invite de commandes ouverte à l'emplacement `c:\temp\CodeDeployGitHubDemo`.

Étape 3 : télécharger un exemple d'application dans votre GitHub référentiel

Au cours de cette étape, vous allez copier un exemple de révision depuis un compartiment Amazon S3 public vers votre GitHub référentiel. (Pour plus de simplicité, les exemples de révisions fournis pour ce didacticiel sont des pages web uniques.)

Note

Si vous utilisez l'une de vos révisions à la place de notre exemple de révision, votre révision doit :

- suivre les instructions des rubriques [Planifier une révision pour CodeDeploy](#) et [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#) ;
- utiliser le type d'instance correspondant ;
- Soyez accessible depuis votre GitHub tableau de bord.

Si votre révision répond à ces exigences, passez à l'[Étape 5 : Création d'une application et d'un groupe de déploiement](#).

Si vous déployez sur une instance Ubuntu Server, vous devrez télécharger dans votre GitHub référentiel une révision compatible avec une instance Ubuntu Server et CodeDeploy. Pour

plus d'informations, consultez [Planifier une révision pour CodeDeploy](#) et [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

Rubriques

- [Envoyer un exemple de révision depuis une machine Linux, macOS ou Unix locale](#)
- [Transmission de type push d'un exemple de révision à partir d'une machine Windows locale](#)

Envoyer un exemple de révision depuis une machine Linux, macOS ou Unix locale

Le terminal étant toujours ouvert, par exemple à l'emplacement /tmp/CodeDeployGitHubDemo, exécutez les commandes suivantes l'une après l'autre :

Note

Si vous prévoyez de le déployer sur une instance Windows Server, SampleApp_Windows.zip remplacez-le par SampleApp_Linux.zip dans les commandes.

(Amazon S3 copy command)

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Où se *(Amazon S3 copy command)* trouve l'un des éléments suivants :

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2` pour la région de l'est des États-Unis (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1` pour la région de l'est des États-Unis (Virginie du Nord)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1` pour la région USA Ouest (Californie du Nord)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2` pour la région de l'ouest des États-Unis (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1` pour la région du Canada (Centre)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1` pour la région Europe (Irlande)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2` pour la région Europe (Londres)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3` pour la région Europe (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1` pour la région Europe (Francfort)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1` pour la région d'Israël (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1` pour la région Asie-Pacifique (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1` pour la région Asie-Pacifique (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2` pour la région Asie-Pacifique (Séoul)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1` pour la région Asie-Pacifique (Singapour)

- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2` pour la région Asie-Pacifique (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4` pour la région Asie-Pacifique (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1` pour la région Asie-Pacifique (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1` pour la région Amérique du Sud (São Paulo)

Transmission de type push d'un exemple de révision à partir d'une machine Windows locale

L'invite de commande étant toujours ouvert, par exemple à l'emplacement `c:\temp\CodeDeployGitHubDemo`, exécutez les commandes suivantes l'une après l'autre :

 Note

Si vous prévoyez de déployer sur une instance Amazon Linux ou RHEL, remplacez-la par `SampleApp_Windows.zip` in dans `SampleApp_Linux.zip` les commandes.

(Amazon S3 copy command)

Décompressez le contenu du fichier zip the directement dans le répertoire local (par exemple `c:\temp\CodeDeployGitHubDemo`), et non dans un nouveau sous-répertoire.

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Où se *(Amazon S3 copy command)* trouve l'un des éléments suivants :

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2` pour la région de l'est des États-Unis (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1` pour la région de l'est des États-Unis (Virginie du Nord)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1` pour la région USA Ouest (Californie du Nord)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2` pour la région de l'ouest des États-Unis (Oregon)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1` pour la région du Canada (Centre)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1` pour la région Europe (Irlande)
- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2` pour la région Europe (Londres)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3` pour la région Europe (Paris)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1` pour la région Europe (Francfort)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1` pour la région d'Israël (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1` pour la région Asie-Pacifique (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1` pour la région Asie-Pacifique (Tokyo)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2` pour la région Asie-Pacifique (Séoul)

- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1` pour la région Asie-Pacifique (Singapour)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2` pour la région Asie-Pacifique (Sydney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4` pour la région Asie-Pacifique (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1` pour la région Asie-Pacifique (Mumbai)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1` pour la région Amérique du Sud (São Paulo)

Pour transférer votre propre révision vers une instance du serveur Ubuntu, copiez-la dans votre dépôt local, puis appelez le système suivant :

```
git add .
git commit -m "Added Ubuntu app"
git push
```

Étape 4 : provisionner une instance

Au cours de cette étape, vous allez créer ou configurer l'instance sur laquelle vous allez déployer l'exemple d'application. Vous pouvez effectuer un déploiement sur une instance Amazon EC2 ou une instance sur site qui exécute l'un des systèmes d'exploitation pris en charge par CodeDeploy. Pour plus d'informations, consultez [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#). (Si une instance est déjà configurée pour être utilisée dans CodeDeploy des déploiements, passez à l'étape suivante.)

Pour provisionner une instance

1. Suivez les instructions [Lancer une instance Amazon EC2 \(console\)](#) pour provisionner une instance.

2. Lorsque vous lancez l'instance, n'oubliez pas de spécifier une balise sur la page Ajouter des balises. Pour plus de détails sur la façon de spécifier la balise, consultez [Lancer une instance Amazon EC2 \(console\)](#).

Pour vérifier que l' CodeDeploy agent est en cours d'exécution sur l'instance

- Suivez les instructions [Vérifiez que l' CodeDeploy agent est en cours d'exécution](#) pour vérifier que l'agent est en cours d'exécution.

Après avoir correctement provisionné l'instance et vérifié que l' CodeDeploy agent est en cours d'exécution, passez à l'étape suivante.

Étape 5 : Création d'une application et d'un groupe de déploiement

Au cours de cette étape, vous allez utiliser la CodeDeploy console ou le AWS CLI pour créer une application et un groupe de déploiement à utiliser pour déployer l'exemple de révision depuis votre GitHub référentiel.

Création d'une application et d'un groupe de déploiement (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Choisissez Créer une application, puis sélectionnez Application personnalisée.
4. Dans Nom de l'application, saisissez **CodeDeployGitHubDemo-App**.
5. Dans Plateforme informatique, choisissez EC2/Sur site.
6. Choisissez Créer une application.
7. Dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement.
8. Dans Nom du groupe de déploiement, saisissez **CodeDeployGitHubDemo-DepGrp**.

9. Dans Rôle de service, choisissez le nom du rôle de CodeDeploy service que vous avez créé dans [Créer un rôle de service pour CodeDeploy](#).
10. Dans Type de déploiement, choisissez Sur place.
11. Dans Configuration de l'environnement, en fonction du type d'instance que vous utilisez, choisissez les instances Amazon EC2 ou les instances sur site. Dans les zones Clé et Valeur, entrez la clé et la valeur de la balise d'instance qui a été appliquée à votre instance à l'[Étape 4 : provisionner une instance](#).
12. Dans Configuration du déploiement, choisissez CodeDeployDefault. AllatOnce.
13. Dans Équilibreur de charge, désactivez Activer l'équilibrage de charge.
14. Développez Avancé.
15. Dans Alarmes, sélectionnez Ignorer les alarmes.
16. Choisissez Créer un groupe de déploiement et passez à l'étape suivante.

Création d'une application et d'un groupe de déploiement (interface de ligne de commande)

1. Appelez la commande create-application pour créer une application dans CodeDeploy nommée CodeDeployGitHubDemo-App :

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Appelez la commande create-deployment-group pour créer un groupe de déploiement nommé CodeDeployGitHubDemo-DepGrp :
 - Si vous effectuez un déploiement sur une instance Amazon EC2, *ec2-tag-key* la clé de balise d'instance Amazon EC2 qui a été appliquée à votre instance Amazon EC2 dans le cadre de. [Étape 4 : provisionner une instance](#)
 - Si vous effectuez un déploiement sur une instance Amazon EC2, *ec2-tag-value* la valeur de balise d'instance Amazon EC2 qui a été appliquée à votre instance Amazon EC2 dans le cadre de. [Étape 4 : provisionner une instance](#)
 - Si vous effectuez un déploiement sur une instance locale, *on-premises-tag-key* la clé de balise d'instance locale qui a été appliquée à votre instance sur site fait-elle partie de. [Étape 4 : provisionner une instance](#)

- Si vous effectuez un déploiement sur une instance locale, *on-premises-tag-value* la valeur de balise d'instance locale qui a été appliquée à votre instance sur site fait-elle partie de. [Étape 4 : provisionner une instance](#)
- *service-role-arn* est l'ARN du rôle de service que vous avez créé dans [Créer un rôle de service pour CodeDeploy](#). (Suivez les instructions de [Obtenez l'ARN du rôle de service \(CLI\)](#) pour trouver le nom ARN du rôle de service.)

```
aws deploy create-deployment-group --application-name CodeDeployGitHubDemo-App
--ec2-tag-filters Key=ec2-tag-key,Type=KEY_AND_VALUE,Value=ec2-tag-value --on-
premises-tag-filters Key=on-premises-tag-key,Type=KEY_AND_VALUE,Value=on-premises-
tag-value --deployment-group-name CodeDeployGitHubDemo-DepGrp --service-role-
arn service-role-arn
```

Note

La [create-deployment-group](#) commande permet de créer des déclencheurs qui entraînent l'envoi de notifications Amazon SNS aux abonnés du topic concernant des événements spécifiques liés aux déploiements et aux instances. La commande prend également en charge des options permettant d'annuler automatiquement les déploiements et de configurer des alarmes pour arrêter les déploiements lorsque les seuils de surveillance des CloudWatch alarmes d'Amazon sont atteints. Les commandes associées à ces actions ne sont pas incluses dans ce didacticiel.

Étape 6 : Déployer l'application sur l'instance

Au cours de cette étape, vous devez utiliser la CodeDeploy console ou le AWS CLI pour déployer l'exemple de révision de votre GitHub référentiel vers votre instance.

Pour déployer la révision (console)

1. Sur la page Détails du groupe de déploiement, choisissez Créer un déploiement.
2. Dans Groupe de déploiement, choisissez **CodeDeployGitHubDemo-DepGrp**.
3. Dans Type de révision, sélectionnez GitHub.
4. Dans Connect to GitHub, effectuez l'une des opérations suivantes :

- Pour créer une connexion entre CodeDeploy les applications et un GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Dans le GitHub compte, entrez un nom pour identifier cette connexion, puis choisissez Connect to GitHub. La page Web vous invite CodeDeploy à autoriser l'interaction avec GitHub l'application nomméeCodeDeployGitHubDemo-App. Passez à l'étape 5.
 - Pour utiliser une connexion que vous avez déjà créée, dans le GitHubcompte, sélectionnez son nom, puis choisissez Connect to GitHub. Passez à l'étape 7.
 - Pour créer une connexion à un autre GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Choisissez Se connecter à un autre GitHub compte, puis sélectionnez Connect to GitHub. Passez à l'étape 5.
5. Suivez les instructions de la page de connexion pour vous connecter avec votre GitHub compte.
 6. Sur la page Authorize application, choisissez Authorize application.
 7. Sur la page CodeDeploy Créer un déploiement, dans Nom du référentiel, entrez le nom GitHub d'utilisateur que vous avez utilisé pour vous connecter, suivi d'une barre oblique (/), suivi du nom du référentiel dans lequel vous avez envoyé la révision de votre application (par exemple,**my-github-user-name/CodeDeployGitHubDemo**).

Si vous n'êtes pas sûr de la valeur à entrer ou si vous souhaitez spécifier un autre référentiel :

- a. Dans un onglet de navigateur Web distinct, accédez à votre [GitHub tableau de bord](#).
- b. Dans Your repositories, placez le pointeur de la souris sur le nom du référentiel cible. Une infobulle apparaît, affichant le nom de l' GitHub utilisateur ou de l'organisation, suivi d'une barre oblique (/), puis du nom du référentiel. Entrez cette valeur dans la zone Nom du référentiel.

 Note

Si le nom du référentiel cible n'apparaît pas dans Vos référentiels, utilisez la GitHub zone de recherche pour trouver le référentiel cible et le nom de GitHub l'utilisateur ou de l'organisation.

8. Dans le champ ID de validation, entrez l'ID du commit associé au transfert de la révision de votre application vers GitHub.

Si vous n'êtes pas sûr de la valeur à entrer :

- a. Dans un onglet de navigateur Web distinct, accédez à votre [GitHub tableau de bord](#).
 - b. Dans Your repositories, choisissez CodeDeployGitHubDemo.
 - c. Dans la liste des validations, recherchez et copiez l'ID de validation associé au push de la révision de votre application vers GitHub. Cet ID contient généralement 40 caractères et se compose de lettres et de chiffres. (N'utilisez pas la version courte de l'ID de validation, qui correspond généralement aux 10 premiers caractères de la version longue.)
 - d. Collez l'ID de validation dans la zone ID de validation.
9. Choisissez Déployer et passez à l'étape suivante.

Pour déployer la révision (interface de ligne de commande)

Avant de pouvoir appeler une AWS CLI commande qui interagit GitHub (telle que la create-deployment commande que vous appellerez ensuite), vous devez CodeDeploy autoriser l'utilisation de votre compte GitHub utilisateur GitHub pour interagir avec l'CodeDeployGitHubDemo-Appapplication. Actuellement, vous devez utiliser la console CodeDeploy pour ce faire.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Choisissez CodeDeployGitHubDemo-App.
4. Sur l'onglet Déploiements, choisissez Créer un déploiement.

Note

Vous n'allez pas créer un nouveau déploiement. C'est actuellement le seul moyen de donner CodeDeploy l'autorisation d'interagir avec vous GitHub au nom de votre compte GitHub utilisateur.

5. Dans le groupe de déploiement, choisissez CodeDeployGitHubDemo- DepGrp.

6. Dans Type de révision, sélectionnez GitHub.
7. Dans Connect to GitHub, effectuez l'une des opérations suivantes :
 - Pour créer une connexion entre CodeDeploy les applications et un GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Dans le GitHub compte, tapez un nom pour identifier cette connexion, puis choisissez Connect to GitHub. La page Web vous invite CodeDeploy à autoriser l'interaction avec GitHub l'application nomméeCodeDeployGitHubDemo-App. Passez à l'étape 8.
 - Pour utiliser une connexion que vous avez déjà créée, dans le GitHubcompte, sélectionnez son nom, puis choisissez Connect to GitHub. Passez à l'étape 10.
 - Pour créer une connexion à un autre GitHub compte, déconnectez-vous GitHub dans un onglet de navigateur Web distinct. Choisissez Se connecter à un autre GitHub compte, puis sélectionnez Connect to GitHub. Passez à l'étape 8.
8. Suivez les instructions de la page de connexion pour vous connecter à l'aide de votre nom GitHub d'utilisateur ou de votre adresse e-mail et de votre mot de passe.
9. Sur la page Authorize application, choisissez Authorize application.
10. Sur la page CodeDeploy Créer un déploiement, choisissez Annuler.
11. Appelez la create-deployment commande pour déployer la révision de votre GitHub référentiel vers l'instance, où :
 - **repository** est le nom de votre GitHub compte, suivi d'une barre oblique (/), suivi du nom de votre dépôt (CodeDeployGitHubDemo), par exemple, MyGitHubUserName/CodeDeployGitHubDemo

Si vous n'êtes pas sûr de la valeur à utiliser ou si vous souhaitez spécifier un autre référentiel :

1. Dans un onglet de navigateur Web distinct, accédez à votre [GitHub tableau de bord](#).
2. Dans Your repositories, placez le pointeur de la souris sur le nom du référentiel cible. Une infobulle apparaît, affichant le nom de l' GitHub utilisateur ou de l'organisation, suivi d'une barre oblique (/), puis du nom du référentiel. Il s'agit de la valeur à utiliser.

 Note

Si le nom du référentiel cible n'apparaît pas dans Vos référentiels, utilisez la GitHub zone de recherche pour trouver le référentiel cible et le nom de GitHub l'utilisateur ou de l'organisation correspondant.

- *commit-id* est le commit associé à la version de la révision de l'application que vous avez envoyée à votre dépôt (par exemple, f835159a...528eb76f).

Si vous n'êtes pas sûr de la valeur à utiliser :

1. Dans un onglet de navigateur Web distinct, accédez à votre [GitHub tableau de bord](#).
2. Dans Your repositories, choisissez CodeDeployGitHubDemo.
3. Dans la liste des validations, trouvez l'ID de validation associé au push de la révision de votre application vers GitHub. Cet ID contient généralement 40 caractères et se compose de lettres et de chiffres. (N'utilisez pas la version courte de l'ID de validation, qui correspond généralement aux 10 premiers caractères de la version longue.) Utilisez cette valeur.

Si vous travaillez sur une machine Linux, macOS ou Unix locale :

```
aws deploy create-deployment \  
  --application-name CodeDeployGitHubDemo-App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name CodeDeployGitHubDemo-DepGrp \  
  --description "My GitHub deployment demo" \  
  --github-location repository=repository,commitId=commit-id
```

Si vous travaillez sur une machine Windows locale :

```
aws deploy create-deployment --application-name CodeDeployGitHubDemo-App --  
deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name  
CodeDeployGitHubDemo-DepGrp --description "My GitHub deployment demo" --github-  
location repository=repository,commitId=commit-id
```

Étape 7 : Surveiller et vérifier le déploiement

Au cours de cette étape, vous allez utiliser la CodeDeploy console ou le AWS CLI pour vérifier le succès du déploiement. Vous devez utiliser votre navigateur web pour afficher la page web déployée sur l'instance que vous avez créée ou configurée.

 Note

Si vous effectuez un déploiement sur une instance Ubuntu Server, utilisez votre propre stratégie de test pour déterminer si la révision déployée fonctionne comme prévu sur l'instance, puis passez à l'étape suivante.

Pour surveiller et vérifier le déploiement (console)

1. Dans le volet de navigation, développez Deploy, puis choisissez Deployments.
2. Dans la liste des déploiements, recherchez la ligne avec une valeur d'application de CodeDeployGitHubDemo-App et une valeur de groupe de déploiement de CodeDeployGitHubDemo -. DepGrp Si ni Succeeded (Réussi), ni Failed (Échec) ne figure dans la colonne Statut, choisissez le bouton Actualiser régulièrement.
3. Si Failed (Échec) s'affiche dans la colonne Statut, suivez les instructions de [Afficher les détails de l'instance \(console\)](#) pour dépanner le déploiement.
4. Si Succeeded (Réussi) s'affiche dans la colonne Statut, vous pouvez maintenant vérifier le déploiement via votre navigateur Web. Notre exemple de révision déploie une page Web unique sur l'instance. Si vous effectuez un déploiement sur une instance Amazon EC2, dans votre navigateur Web, accédez à `http://public-dns` l'instance (par exemple, `http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Si vous pouvez voir la page Web, vous avez réussi la procédure. Maintenant que vous avez réussi AWS CodeDeploy à déployer une révision depuis GitHub, vous pouvez passer à [Étape 8 : nettoyer](#).

Pour surveiller et vérifier le déploiement (interface de ligne de commande)

1. Appelez la commande `list-deployments` pour obtenir l'ID de déploiement de l'application nommée CodeDeployGitHubDemo-App et le groupe de déploiement nommé CodeDeployGitHubDemo-DepGrp :

```
aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --  
deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output  
text
```

2. Appelez la commande `get-deployment`, en fournissant l'ID du déploiement qui figure dans la sortie de la commande `list-deployments` :

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.[status, creator]" --output text
```

3. Si Failed (Échec) est renvoyé, suivez les instructions de [Afficher les détails de l'instance \(console\)](#) pour dépanner le déploiement.
4. Si Succeeded (Réussi) est renvoyé, vous pouvez maintenant essayer de vérifier le déploiement par le biais de votre navigateur Web. Notre exemple de révision correspond à une page Web unique déployée sur l'instance. Si vous effectuez un déploiement sur une instance Amazon EC2, vous pouvez consulter cette page dans votre navigateur Web en accédant à `http://public-dns l'instance Amazon EC2 (par exemple,). http://ec2-01-234-567-890.compute-1.amazonaws.com`
5. Si vous pouvez voir la page Web, vous avez réussi la procédure. Vous avez utilisé avec succès AWS CodeDeploy le déploiement à partir de votre GitHub référentiel.

Étape 8 : nettoyer

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, vous devez mettre fin à l'instance Amazon EC2 et aux ressources associées. Éventuellement, vous pouvez supprimer les enregistrements des composants de déploiement CodeDeploy associés à ce didacticiel. Si vous utilisiez un GitHub dépôt uniquement pour ce didacticiel, vous pouvez également le supprimer maintenant.

Pour supprimer une CloudFormation pile (si vous avez utilisé le CloudFormation modèle pour créer une instance Amazon EC2)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la colonne Piles, choisissez la pile commençant par `CodeDeploySampleStack`.
3. Sélectionnez Delete (Supprimer).
4. À l'invite, choisissez Supprimer la pile. L'instance Amazon EC2 ainsi que le profil d'instance IAM et le rôle de service associés sont supprimés.

Pour annuler l'inscription et nettoyer manuellement une instance sur site (si vous avez mis en service une instance sur site)

1. Utilisez la commande AWS CLI pour appeler la commande de [désenregistrement](#) sur l'instance locale représentée ici par *your-instance-name* et la région associée par : *your-region*

```
aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --region your-region
```

2. À partir de l'instance locale, appelez la commande de [désinstallation](#) :

```
aws deploy uninstall
```

Pour mettre fin manuellement à une instance Amazon EC2 (si vous avez lancé manuellement une instance Amazon EC2)

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
2. Dans le panneau de navigation, sous Instances, choisissez Instances.
3. Cochez la case à côté de l'instance Amazon EC2 que vous souhaitez résilier. Dans le menu Actions, pointez sur État de l'instance, puis choisissez Terminer.
4. À l'invite, choisissez Oui, résilier.

Pour supprimer les enregistrements des composants de CodeDeploy déploiement

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Choisissez CodeDeployGitHubDemo-App.

4. Choisissez Supprimer l'application.
5. Lorsque vous y êtes invité, tapez **Delete**, puis choisissez Supprimer.

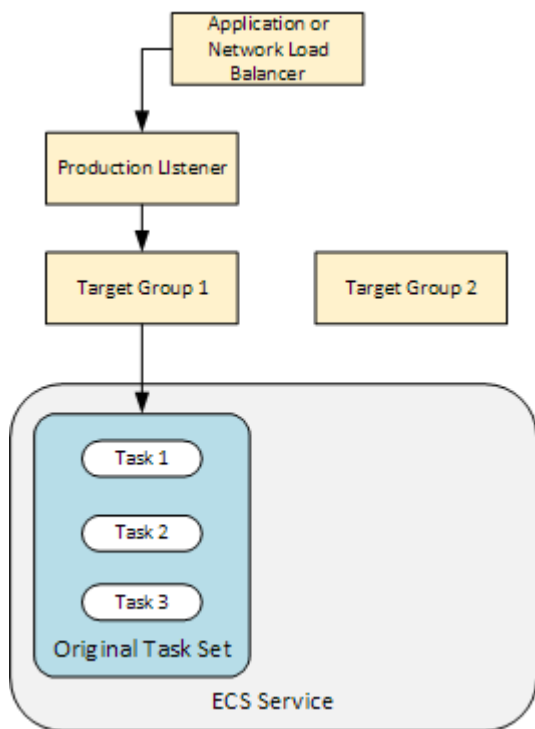
Pour supprimer votre GitHub dépôt

Consultez [la section Suppression d'un dépôt](#) dans [GitHub l'aide](#).

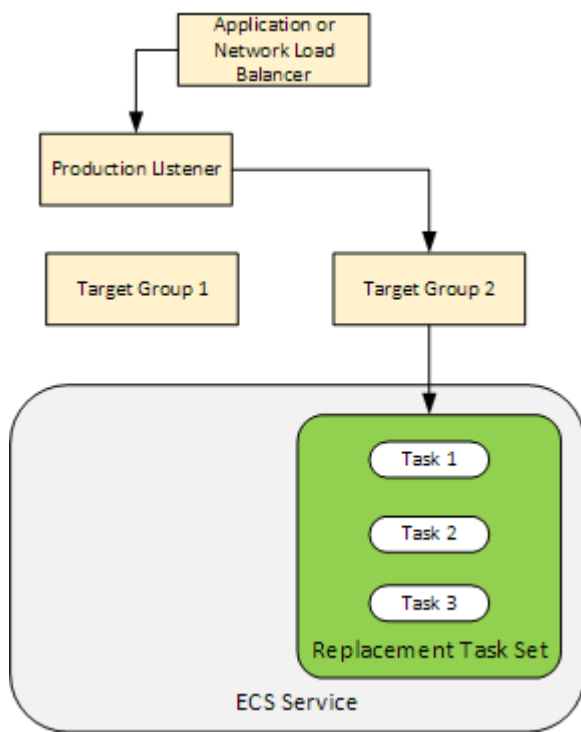
Tutoriel : Déployer une application dans Amazon ECS

Dans ce didacticiel, vous apprendrez à déployer une application dans Amazon ECS à l'aide de CodeDeploy. Vous commencez par une application que vous avez déjà créée et déployée dans Amazon ECS. La première étape consiste à mettre à jour votre application en modifiant son fichier de définition de tâche avec une nouvelle balise. Ensuite, vous devez CodeDeploy déployer la mise à jour. Au cours du déploiement, CodeDeploy installe votre mise à jour dans un nouvel ensemble de tâches de remplacement. Il déplace ensuite le trafic de production de la version d'origine de votre application Amazon ECS, qui se trouve dans son ensemble de tâches d'origine, vers la version mise à jour dans l'ensemble de tâches de remplacement.

Lors d'un déploiement Amazon ECS, CodeDeploy utilise un équilibreur de charge configuré avec deux groupes cibles et un écouteur de trafic de production. Le schéma suivant montre comment l'équilibreur de charge, l'écouteur de production, les groupes cibles et votre application Amazon ECS sont liés avant le début du déploiement. Ce didacticiel utilise un équilibreur de charge Application Load Balancer. Vous pouvez également utiliser un Network Load Balancer.



Après un déploiement réussi, l'écouteur de trafic de production transmet le trafic à votre nouvel ensemble de tâches de remplacement et l'ensemble de tâches d'origine est résilié. Le schéma suivant montre comment vos ressources sont liées après un déploiement réussi. Pour de plus amples informations, veuillez consulter [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#).



Pour plus d'informations sur la façon d'utiliser le AWS CLI pour déployer une application dans Amazon ECS, consultez [Tutoriel : Création d'un service à l'aide d'un blue/green déploiement](#). Pour plus d'informations sur la façon de CodePipeline détecter et de déployer automatiquement les modifications apportées à un service Amazon ECS avec CodeDeploy, consultez [Tutoriel : Création d'un pipeline avec une source Amazon ECR et ECS-to-CodeDeploy déploiement](#).

Une fois ce didacticiel terminé, vous pouvez utiliser l' CodeDeploy application et le groupe de déploiement que vous avez créés pour ajouter un test de validation de déploiement [Tutoriel : Déployer un service Amazon ECS avec un test de validation](#).

Rubriques

- [Conditions préalables](#)
- [Étape 1 : mettez à jour votre application Amazon ECS](#)
- [Étape 2 : Création du AppSpec fichier](#)
- [Étape 3 : utiliser la CodeDeploy console pour déployer votre application](#)
- [Étape 4 : nettoyer](#)

Conditions préalables

Pour effectuer ce didacticiel, vous devez d'abord :

- Effectuer les étapes 2 et 3 de [Commencer avec CodeDeploy](#).
- Créez un Application Load Balancer configuré avec deux groupes cibles et un écouteur. Pour plus d'informations sur la création d'un équilibreur de charge à l'aide de la console, consultez [Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS](#). Pour plus d'informations sur la création d'un équilibreur de charge à l'aide de AWS CLI, consultez [Étape 1 : Création d'un équilibreur de charge d'application](#) dans le guide de l'utilisateur d'Amazon Elastic Container Service. Lorsque vous créez votre équilibreur de charge, notez les informations suivantes pour ce didacticiel :
 - Nom de votre équilibreur de charge.
 - Noms de vos groupes cibles.
 - Port utilisé par l'écouteur de votre équilibreur de charge.
- Créez un cluster et un service Amazon ECS. Pour plus d'informations, consultez les étapes 2, 3 et 4 de la section [Tutoriel : Création d'un service à l'aide d'un blue/green déploiement](#) dans le

guide de l'utilisateur d'Amazon Elastic Container Service. Notez les informations suivantes pour ce didacticiel :

- Le nom de votre cluster Amazon ECS.
- L'ARN de la définition de tâche utilisée par votre service Amazon ECS.
- Le nom du conteneur utilisé par votre service Amazon ECS.
- Créez un compartiment Amazon S3 pour votre AppSpec fichier.

Étape 1 : mettez à jour votre application Amazon ECS

Dans cette section, vous mettez à jour votre application Amazon ECS avec une nouvelle révision de sa définition de tâche. La révision mise à jour ajoute une nouvelle paire clé/balise. Dans [Étape 3 : utiliser la CodeDeploy console pour déployer votre application](#), vous déployez la version mise à jour de votre application Amazon ECS.

Pour mettre à jour votre définition de tâche

1. Ouvrez la console à la <https://console.aws.amazon.com/ecs/version 2>.
2. Dans le panneau de navigation, sélectionnez Task Definitions (Définition des tâches).
3. Choisissez la définition de tâche utilisée par votre service Amazon ECS.
4. Sélectionnez la révision de définition de tâche, puis choisissez Créer une nouvelle révision, Créer une nouvelle révision.
5. Pour ce didacticiel, effectuez une petite mise à jour de la définition de tâche en ajoutant une balise. Au bas de la page, sous Balises, créez une nouvelle balise en saisissant une nouvelle paire clé/valeur.
6. Choisissez Créer.

Le numéro de révision de votre définition de tâche est incrémenté d'une unité.

7. Choisissez l'onglet JSON. Notez ce qui suit, car vous aurez besoin de ces informations à l'étape suivante.
 - La valeur de `taskDefinitionArn`. Son format est `arn:aws:ecs:aws-region:account-id:task-definition/task-definition-family:task-definition-revision`. Il s'agit de l'ARN de votre définition de tâche mise à jour.
 - Dans l'élément `containerDefinitions`, la valeur de `name`. Il s'agit du nom de votre conteneur.

- Dans l'élément `portMappings`, la valeur de `containerPort`. Il s'agit du port de votre conteneur.

Étape 2 : Création du AppSpec fichier

Dans cette section, vous créez votre AppSpec fichier et vous le chargez dans le compartiment Amazon S3 que vous avez créé dans la [Conditions préalables](#) section. Le AppSpec fichier pour un déploiement Amazon ECS indique la définition de votre tâche, le nom du conteneur et le port du conteneur. Pour plus d'informations, consultez [AppSpec Exemple de fichier pour un déploiement Amazon ECS](#) et [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

Pour créer votre AppSpec dossier

1. Si vous souhaitez créer votre AppSpec fichier en utilisant YAML, créez un fichier nommé `appspectest.yaml`. Si vous souhaitez créer votre AppSpec fichier à l'aide de JSON, créez un fichier nommé `appspectest.json`.
2. Choisissez l'onglet approprié, selon que vous utilisez YAML ou JSON pour votre AppSpec fichier, et copiez son contenu dans le AppSpec fichier que vous venez de créer. Pour la propriété `TaskDefinition`, utilisez l'ARN de définition de tâche que vous avez noté dans la section [Étape 1 : mettez à jour votre application Amazon ECS](#).

JSON AppSpec

```
{
  "version": 0.0,
  "Resources": [
    {
      "TargetService": {
        "Type": "AWS::ECS::Service",
        "Properties": {
          "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id:task-definition/ecs-demo-task-definition:revision-number",
          "LoadBalancerInfo": {
            "ContainerName": "your-container-name",
            "ContainerPort": your-container-port
          }
        }
      }
    }
  ]
}
```

```
}
```

YAML AppSpec

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id:task-
definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "your-container-name"
          ContainerPort: your-container-port
```

Note

Votre ensemble de tâches de remplacement hérite du sous-réseau, du groupe de sécurité, de la version de la plateforme et des valeurs IP publiques attribuées de votre ensemble de tâches d'origine. Vous pouvez remplacer ces valeurs pour votre ensemble de tâches de remplacement en définissant leurs propriétés facultatives dans votre AppSpec fichier. Pour plus d'informations, consultez [AppSpec section « ressources » pour les déploiements Amazon ECS](#) et [AppSpec Exemple de fichier pour un déploiement Amazon ECS](#).

3. Importez votre AppSpec fichier dans le compartiment S3 que vous avez créé comme condition préalable à ce didacticiel.

Étape 3 : utiliser la CodeDeploy console pour déployer votre application

Dans cette section, vous allez créer une CodeDeploy application et un groupe de déploiement pour déployer votre application mise à jour dans Amazon ECS. Pendant le déploiement, CodeDeploy déplace le trafic de production de votre application vers sa nouvelle version dans un nouvel ensemble de tâches de remplacement. Pour effectuer cette étape, vous avez besoin des éléments suivants :

- Le nom de votre cluster Amazon ECS.
- Le nom de votre service Amazon ECS.

- Le nom de votre Application Load Balancer.
- Votre port d'écoute de production.
- Noms de vos groupes cibles.
- Nom du compartiment S3 que vous avez créé.

Pour créer une CodeDeploy application

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy/>.
2. Choisissez Créer une application.
3. Dans Nom de l'application, saisissez **ecs-demo-codedeploy-app**.
4. Dans Plateforme de calcul, choisissez Amazon ECS.
5. Choisissez Créer une application.

Pour créer un groupe CodeDeploy de déploiement

1. Dans l'onglet Groupes de déploiement de la page de votre application, choisissez Créer un groupe de déploiement.
2. Dans Nom du groupe de déploiement, saisissez **ecs-demo-dg**.
3. Dans Rôle de service, choisissez un rôle de service qui accorde CodeDeploy l'accès à Amazon ECS. Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès pour AWS CodeDeploy](#).
4. Dans Configuration de l'environnement, choisissez le nom de votre cluster Amazon ECS et le nom du service.
5. Dans Équilibreurs de charge, choisissez le nom de l'équilibreur de charge qui achemine le trafic vers votre service Amazon ECS.
6. Dans Port de l'écouteur de production, choisissez le port et le protocole de l'écouteur qui transmet le trafic de production à votre service Amazon ECS (par exemple, HTTP : 80). Ce didacticiel n'inclut pas d'écouteur de test facultatif. Par conséquent, ne choisissez pas de port dans Port d'écoute de test.
7. À partir de Nom du groupe cible 1 et Nom du groupe cible 2, choisissez deux groupes cibles différents pour acheminer le trafic au cours de votre déploiement. Assurez-vous que ce sont les groupes cibles que vous avez créés pour votre équilibreur de charge. Peu importe lequel est utilisé pour le groupe cible 1 et lequel est utilisé pour le groupe cible 2.

8. Choisissez Réacheminer le trafic immédiatement.
9. Pour Original revision termination (Résiliation de la révision d'origine), choisissez 0 jour, 0 heure et 5 minutes. Cela vous permet de voir votre déploiement se terminer plus rapidement que si vous utilisez la valeur par défaut (1 heure).

Environment configuration

Choose an ECS cluster name

ecs-tutorial-cluster

Choose an ECS service name

ecs-demo-service

Load balancers

Choose a load balancer

ecs-demo-alb

Production listener port

HTTP: 80

Test listener port - *optional*

A test listener is required if you want to test your replacement version before traffic reroutes to it

Target group 1 name

ecs-demo-tg-1

Target group 2 name

ecs-demo-tg-2

Deployment settings

Traffic rerouting

Choose whether traffic reroutes to the replacement environment immediately or waits for you to start the rerouting process

☒ Reroute traffic immediately

☐ Specify when to reroute traffic

Deployment Configuration

CodeDeployDefault.ECSALLAtOnce

Original revision termination

Specify how long CodeDeploy waits before it terminates the original task set. After termination starts, you cannot rollback manually or automatically

Days

0

Hours

0

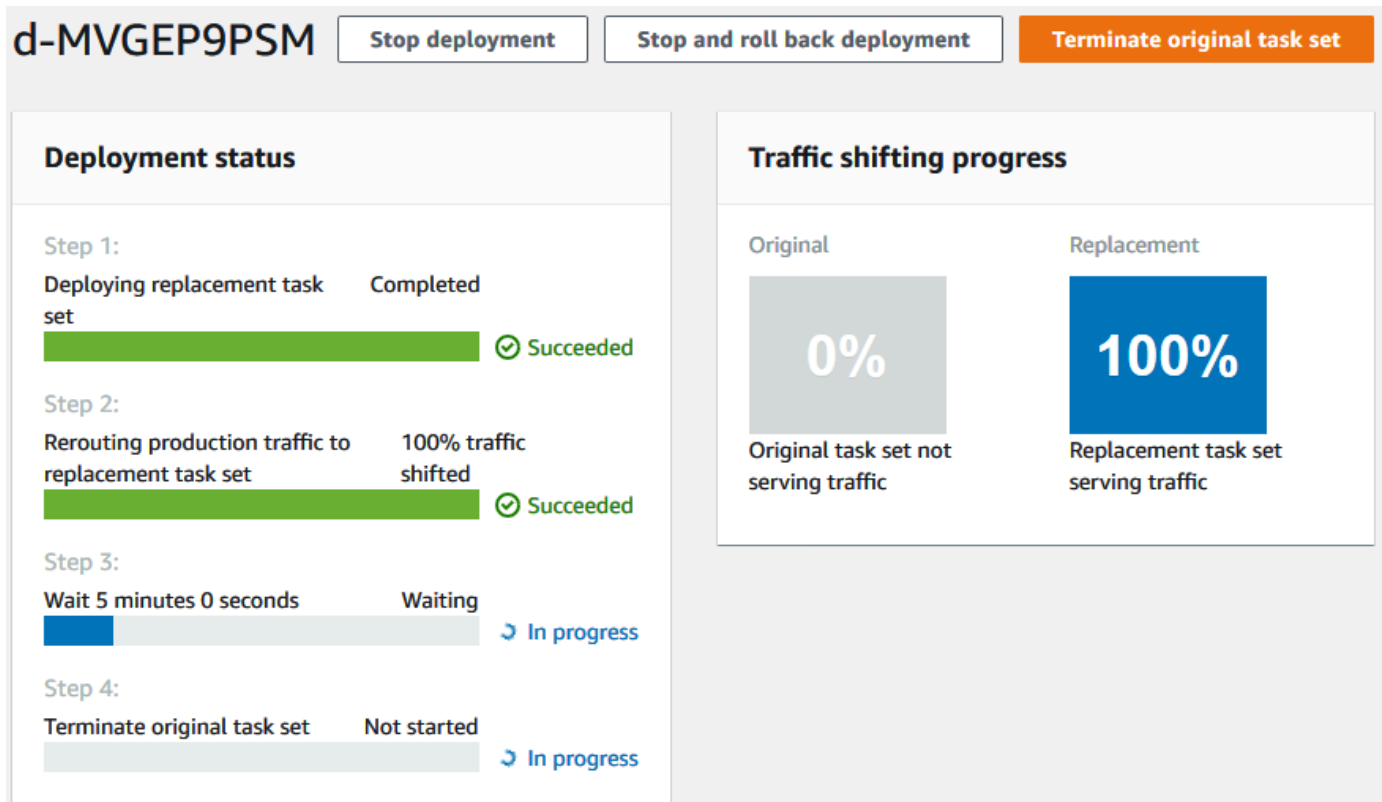
Minutes

5

10. Choisissez Créer un groupe de déploiement.

Pour déployer votre application Amazon ECS

1. Sur la page de la console de votre groupe de déploiement, choisissez Créer un déploiement.
2. Pour le groupe de déploiement, choisissez ecs-demo-dg.
3. Pour Type de révision, choisissez Mon application est stockée dans Amazon S3. Dans Emplacement de la révision, entrez le nom de votre compartiment S3.
4. Pour Type de fichier de révision, choisissez .json ou .yaml, selon le cas.
5. (Facultatif) Dans la zone Description du déploiement, entrez une description de votre déploiement.
6. Choisissez Créer un déploiement.
7. Dans Deployment status (Statut du déploiement), vous pouvez surveiller votre déploiement. Une fois que 100 % du trafic de production est acheminé vers l'ensemble de tâches de remplacement et avant l'expiration du délai d'attente de cinq minutes, vous pouvez choisir Terminer le jeu de tâches d'origine pour mettre fin immédiatement à l'ensemble de tâches d'origine. Si vous ne choisissez pas Terminate original task set (Résilier l'ensemble de tâches d'origine), l'ensemble de tâches d'origine est résilié après l'expiration du délai d'attente de cinq minutes que vous avez spécifié.



Étape 4 : nettoyer

Le didacticiel suivant, [Tutoriel : Déployer un service Amazon ECS avec un test de validation](#), repose sur ce didacticiel et utilise l'application CodeDeploy et le groupe de déploiement que vous avez créés. Si vous souhaitez suivre les étapes de ce didacticiel, ignorez cette étape et ne supprimez pas les ressources que vous avez créées.

Note

Les CodeDeploy ressources que vous avez créées ne sont pas facturées sur votre AWS compte.

Les noms des ressources utilisés dans ces étapes sont ceux suggérés dans ce didacticiel (par exemple, **ecs-demo-codedeploy-app** pour le nom de votre CodeDeploy application). Si vous avez utilisé d'autres noms, veuillez à les utiliser lors du nettoyage.

1. Utilisez la [delete-deployment-group](#) commande pour supprimer le groupe CodeDeploy de déploiement.

```
aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --  
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Utilisez la commande [delete-application](#) pour supprimer l' CodeDeploy application.

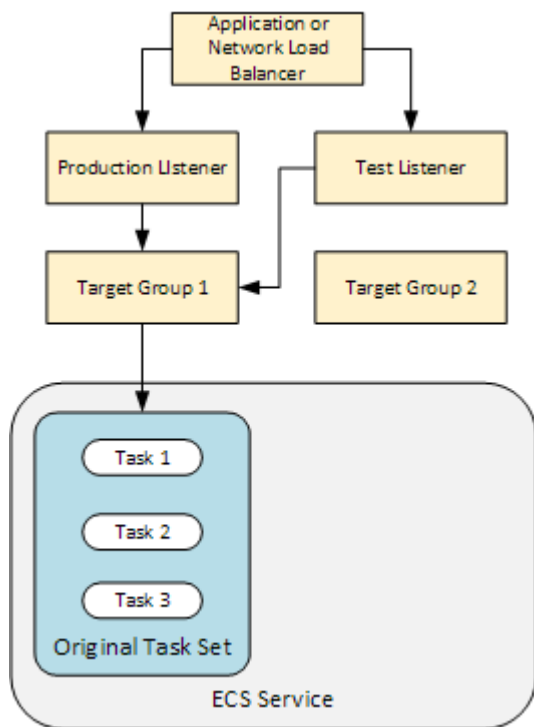
```
aws deploy delete-application --application-name ecs-demo-codedeploy-app --  
region aws-region-id
```

Tutoriel : Déployer un service Amazon ECS avec un test de validation

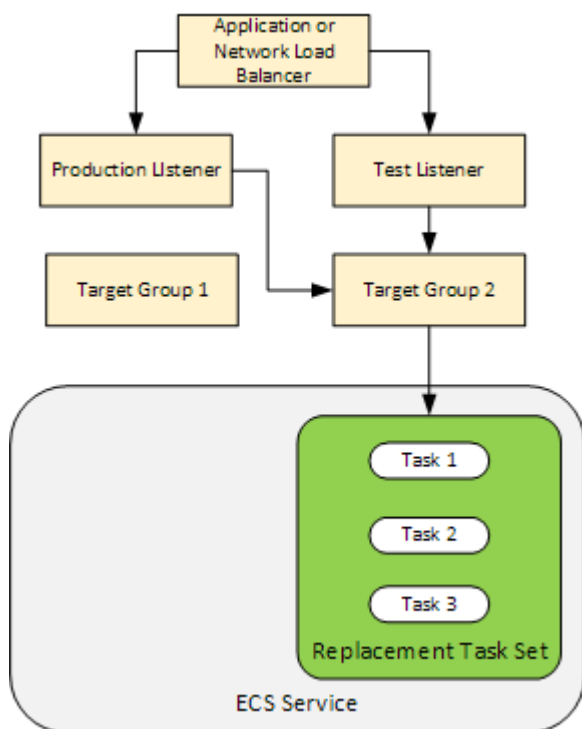
Dans ce didacticiel, vous apprendrez à utiliser une fonction Lambda pour valider une partie du déploiement d'une application Amazon ECS mise à jour. Ce didacticiel utilise l' CodeDeploy application, CodeDeploy le groupe de déploiement et l'application Amazon ECS que vous avez utilisés [Tutoriel : Déployer une application dans Amazon ECS](#). Suivez ce didacticiel avant de commencer celui-ci.

Pour ajouter un test de validation, vous devez d'abord implémenter le test dans une fonction Lambda. Ensuite, dans votre AppSpec fichier de déploiement, vous spécifiez la fonction Lambda pour le hook du cycle de vie que vous souhaitez tester. Si un test de validation échoue, le déploiement s'arrête, est annulé et marqué comme ayant échoué. Si le test réussit, le déploiement se poursuit jusqu'au prochain hook ou événement de cycle de vie de déploiement.

Lors d'un déploiement Amazon ECS avec tests de validation, CodeDeploy utilise un équilibreur de charge configuré avec deux groupes cibles : un écouteur de trafic de production et un écouteur de trafic de test. Le schéma suivant montre comment l'équilibreur de charge, les écouteurs de production et de test, les groupes cibles et votre application Amazon ECS sont liés avant le début du déploiement. Ce didacticiel utilise un équilibreur de charge Application Load Balancer. Vous pouvez également utiliser un Network Load Balancer.



Lors d'un déploiement d'Amazon ECS, il existe cinq points de cycle de vie à tester. Ce didacticiel implémente un test au cours du troisième hook de déploiement du cycle de vie, `AfterAllowTestTraffic`. Pour de plus amples informations, veuillez consulter [Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS](#). Après un déploiement réussi, l'écouteur de trafic de production transmet le trafic à votre nouvel ensemble de tâches de remplacement et l'ensemble de tâches d'origine est résilié. Le schéma suivant montre comment vos ressources sont liées après un déploiement réussi. Pour de plus amples informations, veuillez consulter [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#).



Note

L'exécution de ce didacticiel peut entraîner des frais sur votre AWS compte. Cela inclut les frais éventuels pour CodeDeploy AWS Lambda, et CloudWatch. Pour plus d'informations, consultez les [AWS CodeDeploy sections AWS Lambda Tarification, Tarification](#) et [CloudWatch Tarification Amazon](#).

Rubriques

- [Conditions préalables](#)
- [Étape 1 : créer un écouteur de test](#)
- [Étape 2 : mettez à jour votre application Amazon ECS](#)
- [Étape 3 : Création d'une fonction Lambda Lifecycle Hook](#)
- [Étape 4 : mettez à jour votre AppSpec fichier](#)
- [Étape 5 : utiliser la CodeDeploy console pour déployer votre service Amazon ECS](#)
- [Étape 6 : Afficher la sortie de votre fonction Lambda hook dans Logs CloudWatch](#)
- [Étape 7 : nettoyer](#)

Conditions préalables

Pour réussir ce didacticiel, vous devez d'abord :

- Respecter les conditions préalables énoncées dans [Conditions préalables](#) pour [Tutoriel : Déployer une application dans Amazon ECS](#).
- Suivez les étapes de [Tutoriel : Déployer une application dans Amazon ECS](#). Prenez en compte les informations suivantes :
 - Nom de votre équilibreur de charge.
 - Noms de vos groupes cibles.
 - Port utilisé par l'écouteur de votre équilibreur de charge.
 - ARN de votre équilibreur de charge. Vous l'utilisez pour créer un nouvel écouteur.
 - ARN de l'un de vos groupes cibles. Vous l'utilisez pour créer un nouvel écouteur.
 - L' CodeDeploy application et le groupe de déploiement que vous créez.
 - Le AppSpec fichier que vous créez et qui est utilisé par votre CodeDeploy déploiement. Vous allez modifier ce fichier dans ce didacticiel.

Étape 1 : créer un écouteur de test

Un déploiement Amazon ECS avec des tests de validation nécessite un deuxième écouteur. Cet écouteur est utilisé pour acheminer le trafic de test vers votre application Amazon ECS mise à jour dans le cadre d'un ensemble de tâches de remplacement. Vos tests de validation s'exécutent sur le trafic de test.

L'écouteur de votre trafic de test peut utiliser l'un ou l'autre de vos groupes cibles. Utilisez la AWS CLI commande [create-listener](#) pour créer un deuxième écouteur avec une règle par défaut qui transmet le trafic de test vers le port 8080. Utilisez l'ARN de votre équilibreur de charge et l'ARN de l'un de vos groupes cibles.

```
aws elbv2 create-listener --load-balancer-arn your-load-balancer-arn \  
--protocol HTTP --port 8080 \  
--default-actions Type=forward,TargetGroupArn=your-target-group-arn --region your-aws-region
```

Étape 2 : mettez à jour votre application Amazon ECS

Dans cette section, vous allez mettre à jour votre application Amazon ECS afin d'utiliser une nouvelle version de sa définition de tâche. Vous créez la nouvelle révision et y ajoutez une mise à jour mineure en ajoutant une balise.

Pour mettre à jour votre définition de tâche

1. Ouvrez la console Amazon ECS Classic à l'adresse <https://console.aws.amazon.com/ecs/>.
2. Dans le panneau de navigation, sélectionnez Task Definitions (Définition des tâches).
3. Cochez la case correspondant à la définition de tâche utilisée par votre service Amazon ECS.
4. Choisissez Créer une révision.
5. Effectuez une petite mise à jour de la définition de tâche en ajoutant une balise. En bas de la page, dans Balises, créez une nouvelle balise en entrant une nouvelle paire clé/valeur.
6. Choisissez Créer. Vous devriez voir que le numéro de révision de votre définition de tâche a été incrémenté de un.
7. Choisissez l'onglet JSON. Notez la valeur de taskDefinitionArn. Son format est `arn:aws:ecs:aws-region:account-id:task-definition/task-definition-family:task-definition-revision`. Il s'agit de l'ARN de votre définition de tâche mise à jour.

Étape 3 : Création d'une fonction Lambda Lifecycle Hook

Dans cette section, vous implémentez une fonction Lambda pour le hook de votre déploiement Amazon ECS. `AfterAllowTestTraffic` La fonction Lambda exécute un test de validation avant l'installation de l'application Amazon ECS mise à jour. Pour ce didacticiel, la fonction Lambda est renvoyée. Succeeded Lors d'un déploiement réel, les tests de validation renvoient Succeeded ou Failed, en fonction du résultat du test de validation. Lors d'un déploiement réel, vous pouvez également implémenter une fonction de test Lambda pour un ou plusieurs autres hooks d'événements du cycle de vie du déploiement d'Amazon ECS (`BeforeInstall`, `AfterInstallBeforeAllowTraffic`, et `AfterAllowTraffic`). Pour de plus amples informations, veuillez consulter [Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS](#).

Un rôle IAM est requis pour créer votre fonction Lambda. Le rôle accorde à la fonction Lambda l'autorisation d'écrire dans CloudWatch Logs et de définir l'état d'un hook du CodeDeploy cycle de vie.

Pour créer un rôle IAM

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Sélectionnez Rôles dans le volet de navigation, puis Créer un rôle.
3. Créez un rôle avec les propriétés suivantes :
 - Entité de confiance : AWS Lambda.
 - Autorisations : AWSLambdaBasicExecutionRole. Cela donne à votre fonction Lambda l'autorisation d'écrire dans Logs. CloudWatch
 - Nom de rôle : **lambda-cli-hook-role**.

Pour plus d'informations, consultez la section [Création d'un rôle AWS Lambda d'exécution](#).

4. Attachez l'autorisation `codedeploy:PutLifecycleEventHookExecutionStatus` au rôle que vous avez créé. Cela donne à vos fonctions Lambda l'autorisation de définir le statut d'un hook de CodeDeploy cycle de vie lors d'un déploiement. Pour plus d'informations, consultez la section [Ajout d'autorisations d'identité IAM](#) dans le Guide de l'Gestion des identités et des accès AWS utilisateur et [PutLifecycleEventHookExecutionStatus](#) dans le Guide de référence des CodeDeploy API.

Pour créer une fonction Lambda de type **AfterAllowTestTraffic** hook

1. Créez un fichier nommé `AfterAllowTestTraffic.js` avec les contenus suivants.

```
'use strict';

const AWS = require('aws-sdk');
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {

  console.log("Entering AfterAllowTestTraffic hook.");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event payload
  var deploymentId = event.DeploymentId;
```

```
var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
var validationTestResult = "Failed";

// Perform AfterAllowTestTraffic validation tests here. Set the test result
// to "Succeeded" for this tutorial.
console.log("This is where AfterAllowTestTraffic validation tests happen.")
validationTestResult = "Succeeded";

// Complete the AfterAllowTestTraffic hook by sending CodeDeploy the validation
status
var params = {
  deploymentId: deploymentId,
  lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
  status: validationTestResult // status can be 'Succeeded' or 'Failed'
};

// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
  if (err) {
    // Validation failed.
    console.log('AfterAllowTestTraffic validation tests failed');
    console.log(err, err.stack);
    callback("CodeDeploy Status update failed");
  } else {
    // Validation succeeded.
    console.log("AfterAllowTestTraffic validation tests succeeded");
    callback(null, "AfterAllowTestTraffic validation tests succeeded");
  }
});
}
```

2. Créez un package de déploiement Lambda.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Utilisez la `create-function` commande pour créer une fonction Lambda pour votre `AfterAllowTestTraffic` hook.

```
aws lambda create-function --function-name AfterAllowTestTraffic \
  --zip-file fileb://AfterAllowTestTraffic.zip \
  --handler AfterAllowTestTraffic.handler \
  --runtime nodejs10.x \
```

```
--role arn:aws:iam::aws-account-id:role/lambda-cli-hook-role
```

4. Notez l'ARN de votre fonction Lambda dans la `create-function` réponse. Vous utilisez cet ARN lorsque vous mettez à jour le AppSpec fichier de votre CodeDeploy déploiement à l'étape suivante.

Étape 4 : mettez à jour votre AppSpec fichier

Dans cette section, vous mettez à jour votre AppSpec fichier avec une Hooks section. Dans Hooks cette section, vous spécifiez une fonction Lambda pour le hook du `AfterAllowTestTraffic` cycle de vie.

Pour mettre à jour votre AppSpec dossier

1. Ouvrez le AppSpec fichier que vous avez créé dans [Étape 2 : Création du AppSpec fichier](#) le [Tutoriel : Déployer une application dans Amazon ECS](#).
2. Mettez à jour la propriété `TaskDefinition` avec l'ARN de définition de tâche que vous avez noté dans [Étape 2 : mettez à jour votre application Amazon ECS](#).
3. Copiez et collez la Hooks section dans votre AppSpec fichier. Mettez ensuite à jour l'ARN `AfterAllowTestTraffic` avec l'ARN de la fonction Lambda que vous avez noté dans. [Étape 3 : Création d'une fonction Lambda Lifecycle Hook](#)

JSON AppSpec

```
{
  "version": 0.0,
  "Resources": [
    {
      "TargetService": {
        "Type": "AWS::ECS::Service",
        "Properties": {
          "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id::task-definition/ecs-demo-task-definition:revision-number",
          "LoadBalancerInfo": {
            "ContainerName": "sample-website",
            "ContainerPort": 80
          }
        }
      }
    }
  ]
}
```

```
    }
  ],
  "Hooks": [
    {
      "AfterAllowTestTraffic": "arn:aws:lambda:aws-region-id:aws-account-id:function:AfterAllowTestTraffic"
    }
  ]
}
```

YAML AppSpec

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id::task-definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "sample-website"
          ContainerPort: 80
      Hooks:
        - AfterAllowTestTraffic: "arn:aws:lambda:aws-region-id:aws-account-id:function:AfterAllowTestTraffic"
```

4. Enregistrez votre AppSpec fichier et chargez-le dans son compartiment S3.

Étape 5 : utiliser la CodeDeploy console pour déployer votre service Amazon ECS

Dans cette section, vous mettez à jour votre groupe de déploiement en spécifiant le port de votre écouteur de test. Il s'agit de l'écouteur que vous avez créé dans [Étape 1 : créer un écouteur de test](#). Pendant le déploiement, CodeDeploy exécute votre test de validation au cours du hook de cycle de vie de déploiement `AfterAllowTestTraffic` en utilisant le trafic de test transmis à votre ensemble de tâches de remplacement à l'aide de l'écouteur de test. Comme votre test de validation renvoie le résultat `Succeeded`, le déploiement se poursuit avec l'événement de cycle de vie de déploiement suivant. Dans un scénario réel, votre fonction de test renvoie `Succeeded` ou `Failed`.

Pour ajouter un écouteur de test à votre groupe de déploiement

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy/>.
2. Dans le volet de navigation, choisissez Applications.
3. Choisissez l'application que vous avez créée dans [Tutoriel : Déployer une application dans Amazon ECS](#). Si vous avez utilisé le nom suggéré, il s'agit de ecs-demo-codedeploy-app.
4. Dans Groupes de déploiement, choisissez le groupe de déploiement que vous avez créé dans [Tutoriel : Déployer une application dans Amazon ECS](#). Si vous avez utilisé le nom suggéré, il s'agit de ecs-demo-dg.
5. Choisissez Modifier.
6. Dans Port d'écoute de test, choisissez le port et le protocole de l'écouteur de test que vous avez créé précédemment dans ce didacticiel. Il doit s'agir de HTTP : 8080.
7. Sélectionnez Enregistrer les modifications.

Pour déployer votre application Amazon ECS

1. Sur la page de la console de votre groupe de déploiement, choisissez Créer un déploiement.
2. Pour le groupe de déploiement, choisissez ecs-demo-dg.
3. Pour Type de révision, choisissez Mon application est stockée dans Amazon S3. Dans Emplacement de la révision, entrez le nom de votre compartiment et de votre AppSpec fichier S3 (par exemple, **s3://my-s3-bucket/appspec.json**).
4. Pour Type de fichier de révision, choisissez .json ou .yaml selon le cas.
5. (Facultatif) Dans la zone Description du déploiement, entrez une description de votre déploiement.
6. Choisissez Créer un déploiement.

Vous pouvez surveiller votre déploiement dans Deployment status (Statut du déploiement). Une fois que 100 % du trafic de production est acheminé vers l'ensemble de tâches de remplacement, vous pouvez choisir Terminer le jeu de tâches d'origine pour mettre fin immédiatement à l'ensemble de tâches d'origine. Si vous ne choisissez pas Terminate original task set (Résilier l'ensemble de tâches d'origine), l'ensemble de tâches d'origine est résilié après la durée que vous avez spécifiée lorsque vous avez créé votre groupe de déploiement.

The screenshot displays the AWS CodeDeploy console interface. At the top, there are three buttons: "Stop deployment", "Stop and roll back deployment", and "Terminate original task set". Below these, the "Deployment status" section shows five steps:

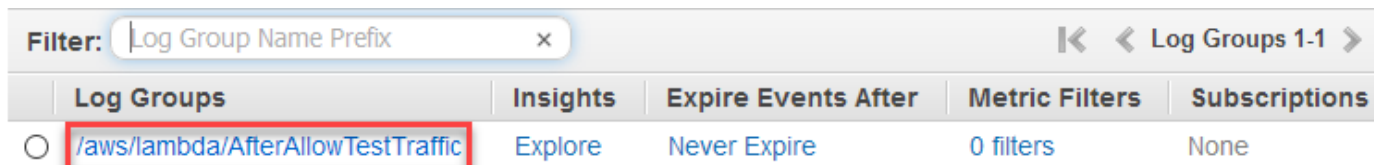
- Step 1:** Deploying replacement task set. Status: Completed. Progress bar is green. Status: Succeeded.
- Step 2:** Test traffic route setup. Status: Completed. Progress bar is green. Status: Succeeded.
- Step 3:** Rerouting production traffic to replacement task set. Status: 100% traffic shifted. Progress bar is green. Status: Succeeded.
- Step 4:** Wait 5 minutes 0 seconds. Status: Waiting. Progress bar is blue. Status: In progress.
- Step 5:** Terminate original task set. Status: Not started. Progress bar is grey. Status: In progress.

The "Traffic shifting progress" section shows two columns: "Original" and "Replacement". The "Original" column shows a grey box with "0%" and the text "Original task set not serving traffic". The "Replacement" column shows a blue box with "100%" and the text "Replacement task set serving traffic".

Étape 6 : Afficher la sortie de votre fonction Lambda hook dans Logs CloudWatch

Si votre CodeDeploy déploiement est réussi, les tests de validation de vos fonctions de hook Lambda sont également réussis. Vous pouvez le confirmer en consultant le journal de la fonction hook dans CloudWatch Logs.

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Logs (Journaux). Vous devriez voir un nouveau groupe de journaux pour la fonction de crochet Lambda que vous avez spécifiée dans votre AppSpec fichier.



3. Choisissez le nouveau groupe de journaux. Cela devrait être `/aws/lambda/AfterAllowTestTrafficHook`.
4. Choisissez le flux de journaux. Si vous voyez plusieurs flux de journaux, choisissez celui dont la date et l'heure sont les plus récentes sous Heure du dernier événement.
5. Développez les événements du flux du journal pour confirmer que votre fonction Lambda Hook a écrit des messages de réussite dans le journal. Ce qui suit montre que la fonction de crochet `AfterAllowTraffic` Lambda a réussi.

Time (UTC +00:00)	Message
2019-09-11	
	No older events
20:11:20	START RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Version: \$LATEST
	START RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Version: \$LATEST
20:11:21	2019-09-11T20:11:21.033Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO Entering AfterAllowTestTraffic hook.
	2019-09-11T20:11:21.033Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO Entering AfterAllowTestTraffic hook.
20:11:21	2019-09-11T20:11:21.034Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO This is where AfterAllowTestTraffic validation tests happen.
	2019-09-11T20:11:21.034Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO This is where AfterAllowTestTraffic validation tests happen.
20:11:21	2019-09-11T20:11:21.789Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO AfterAllowTestTraffic validation tests succeeded
	2019-09-11T20:11:21.789Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO AfterAllowTestTraffic validation tests succeeded
20:11:21	END RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916
20:11:21	REPORT RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Duration: 977.80 ms Billed Duration: 1000 ms Memory Size: 128 MB Ma

Étape 7 : nettoyer

Lorsque vous avez terminé ce didacticiel, nettoyez les ressources qui lui sont associées afin d'éviter la facturation de frais pour des ressources que vous n'utilisez pas. Les noms des ressources utilisés dans cette étape sont ceux proposés dans ce didacticiel (par exemple, **ecs-demo-codedeploy-app** pour le nom de votre CodeDeploy application). Si vous avez utilisé d'autres noms, veillez à les utiliser lors du nettoyage.

Pour nettoyer les ressources du didacticiel

1. Utilisez la [delete-deployment-group](#) commande pour supprimer le groupe CodeDeploy de déploiement.

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --  
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Utilisez la commande [delete-application](#) pour supprimer l' CodeDeploy application.

```
aws deploy delete-application --application-name ecs-demo-deployment-group --  
region aws-region-id
```

3. Utilisez la commande [delete-function](#) pour supprimer votre fonction de crochet Lambda.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

4. Utilisez la [delete-log-group](#) commande pour supprimer votre groupe de CloudWatch journaux.

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutoriel : Déployer une fonction Lambda mise à jour avec CodeDeploy et le modèle d'application AWS sans serveur

AWS SAM est un framework open source permettant de créer des applications sans serveur. Il transforme et étend la syntaxe YAML d'un AWS SAM modèle en CloudFormation syntaxe permettant de créer des applications sans serveur, telles qu'une fonction Lambda. Pour plus d'informations, voir [Qu'est-ce que le modèle d'application AWS sans serveur ?](#)

Dans ce didacticiel, vous allez utiliser AWS SAM pour créer une solution qui effectue les opérations suivantes :

- Crée votre fonction Lambda.
- Crée votre CodeDeploy application et votre groupe de déploiement.
- Crée deux fonctions Lambda qui exécutent des tests de validation du déploiement pendant les hooks CodeDeploy du cycle de vie.
- Détecte la mise à jour de votre fonction Lambda. La mise à jour de la fonction Lambda déclenche un déploiement CodeDeploy qui déplace progressivement le trafic de production de la version d'origine de votre fonction Lambda vers la version mise à jour.

 Note

Ce didacticiel nécessite que vous créiez des ressources qui peuvent entraîner des frais sur votre compte AWS . Cela inclut les frais éventuels pour CodeDeploy, Amazon CloudWatch, et AWS Lambda. Pour plus d'informations, consultez les [CodeDeploy sections Tarification](#), [CloudWatch Tarification Amazon](#) et [AWS Lambda Tarification](#).

Rubriques

- [Conditions préalables](#)
- [Étape 1 : configurer votre infrastructure](#)
- [Étape 2 : mise à jour de la fonction Lambda](#)
- [Étape 3 : Déploiement de la fonction Lambda mise à jour](#)
- [Étape 4 : Afficher les résultats de votre déploiement](#)
- [Étape 5 : nettoyer](#)

Conditions préalables

Pour effectuer ce didacticiel, vous devez d'abord :

- Suivez les étapes de [Commencer avec CodeDeploy](#).
- Installez la AWS Serverless Application Model CLI. Pour plus d'informations, consultez la section [Installation de la CLI AWS SAM](#).
- Créez un compartiment S3. AWS SAM télécharge les artefacts référencés dans votre [modèle AWS SAM](#) dans ce compartiment.

Étape 1 : configurer votre infrastructure

Cette rubrique explique comment créer des fichiers AWS SAM pour votre AWS SAM modèle et vos fonctions Lambda. Ensuite, vous utilisez les `deploy` commandes AWS SAM package et pour générer les composants de votre infrastructure. Lorsque votre infrastructure est prête, vous disposez d'une CodeDeploy application et d'un groupe de déploiement, de la fonction Lambda à mettre à jour et à déployer, et de deux fonctions Lambda contenant des tests de validation exécutés lorsque vous déployez la fonction Lambda. Lorsque vous avez terminé, vous pouvez l'utiliser CloudFormation pour afficher vos composants dans la console Lambda ou AWS CLI pour tester votre fonction Lambda.

Rubriques

- [Créez vos fichiers](#)
- [Package de l'application AWS SAM](#)
- [Déployer l'application AWS SAM](#)
- [\(Facultatif\) inspectez et testez votre infrastructure](#)

Créez vos fichiers

Pour créer votre infrastructure, vous devez créer les fichiers suivants :

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Rubriques

- [Créez votre modèle AWS SAM](#)
- [Créez un fichier pour votre fonction Lambda](#)
- [Créez un fichier pour votre fonction BeforeAllowTraffic Lambda](#)
- [Créez un fichier pour votre fonction AfterAllowTraffic Lambda](#)

Créez votre modèle AWS SAM

Créez un fichier modèle AWS SAM qui spécifie les composants de votre infrastructure.

Pour créer votre modèle AWS SAM

1. Créez un répertoire nommé `SAM-Tutorial`.
2. Dans votre répertoire `SAM-Tutorial`, créez un fichier nommé `template.yml`.
3. Copiez le code YAML suivant dans `template.yml`. Il s'agit de votre modèle AWS SAM .

```
AWSTemplateFormatVersion : '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
Description: A sample SAM template for deploying Lambda functions.
```

```

Resources:
# Details about the myDateTimeFunction Lambda function
myDateTimeFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: myDateTimeFunction.handler
    Runtime: nodejs18.x
# Instructs your myDateTimeFunction is published to an alias named "live".
  AutoPublishAlias: live
# Grants this function permission to call lambda:InvokeFunction
  Policies:
    - Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Action:
            - "lambda:InvokeFunction"
          Resource: '*'
    DeploymentPreference:
# Specifies the deployment configuration
      Type: Linear10PercentEvery1Minute
# Specifies Lambda functions for deployment lifecycle hooks
      Hooks:
        PreTraffic: !Ref beforeAllowTraffic
        PostTraffic: !Ref afterAllowTraffic

# Specifies the BeforeAllowTraffic lifecycle hook Lambda function
beforeAllowTraffic:
  Type: AWS::Serverless::Function
  Properties:
    Handler: beforeAllowTraffic.handler
    Policies:
      - Version: "2012-10-17"
# Grants this function permission to call
      codedeploy:PutLifecycleEventHookExecutionStatus
      Statement:
        - Effect: "Allow"
          Action:
            - "codedeploy:PutLifecycleEventHookExecutionStatus"
          Resource:
            !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
      - Version: "2012-10-17"
# Grants this function permission to call lambda:InvokeFunction
      Statement:

```

```

    - Effect: "Allow"
      Action:
        - "lambda:InvokeFunction"
      Resource: !Ref myDateTimeFunction.Version
      Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
      FunctionName: 'CodeDeployHook_beforeAllowTraffic'
      DeploymentPreference:
        Enabled: false
      Timeout: 5
      Environment:
        Variables:
          NewVersion: !Ref myDateTimeFunction.Version

# Specifies the AfterAllowTraffic lifecycle hook Lambda function
      afterAllowTraffic:
        Type: AWS::Serverless::Function
        Properties:
          Handler: afterAllowTraffic.handler
          Policies:
            - Version: "2012-10-17"
              Statement:
# Grants this function permission to call
                codedeploy:PutLifecycleEventHookExecutionStatus
            - Effect: "Allow"
              Action:
                - "codedeploy:PutLifecycleEventHookExecutionStatus"
              Resource:
                !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
            - Version: "2012-10-17"
              Statement:
# Grants this function permission to call lambda:InvokeFunction
            - Effect: "Allow"
              Action:
                - "lambda:InvokeFunction"
              Resource: !Ref myDateTimeFunction.Version
          Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
          FunctionName: 'CodeDeployHook_afterAllowTraffic'
          DeploymentPreference:
            Enabled: false
          Timeout: 5
          Environment:

```



```
Variables:  
  NewVersion: !Ref myDateTimeFunction.Version
```

Ce modèle spécifie les éléments suivants. Pour plus d'informations, consultez la section [Concepts des AWS SAM modèles](#).

Une fonction Lambda appelée **myDateTimeFunction**

Lorsque cette fonction Lambda est publiée, la `AutoPublishAlias` ligne du modèle la lie à un alias nommé. `live` Plus loin dans ce didacticiel, une mise à jour de cette fonction déclenche un déploiement AWS CodeDeploy qui déplace progressivement le trafic de production de la version d'origine vers la version mise à jour.

Deux fonctions de validation du déploiement Lambda

Les fonctions Lambda suivantes sont exécutées pendant les hooks CodeDeploy du cycle de vie. Ces fonctions contiennent du code qui valide le déploiement de la fonction `myDateTimeFunction` mise à jour. Le résultat des tests de validation est transmis à CodeDeploy à l'aide de sa méthode d'API `PutLifecycleEventHookExecutionStatus`. Si un test de validation échoue, le déploiement échoue et est annulé.

- `CodeDeployHook_beforeAllowTraffic` s'exécute pendant le hook `BeforeAllowTraffic`.
- `CodeDeployHook_afterAllowTraffic` s'exécute pendant le hook `AfterAllowTraffic`.

Le nom des deux fonctions commence par `CodeDeployHook_`. Le `CodeDeployRoleForLambda` rôle autorise les appels à la `invoke` méthode Lambda uniquement dans les fonctions Lambda dont le nom commence par ce préfixe. Pour plus d'informations, veuillez consulter les sections [AppSpec section « hooks » pour un déploiement AWS Lambda](#) et [PutLifecycleEventHookExecutionStatus](#) (français non garanti) de la Référence d'API CodeDeploy .

Détection automatique d'une fonction Lambda mise à jour

Le terme `AutoPublishAlias` indique au cadre de détecter quand la fonction `myDateTimeFunction` change, puis de la déployer à l'aide de l'alias `live`.

Une configuration de déploiement

La configuration de déploiement détermine la vitesse à laquelle votre CodeDeploy application déplace le trafic de la version d'origine de la fonction Lambda vers la nouvelle version. Ce modèle spécifie la configuration de déploiement prédéfinie `Linear10PercentEvery1Minute`.

 Note

Vous ne pouvez pas spécifier de configuration de déploiement personnalisée dans un modèle AWS SAM. Pour de plus amples informations, veuillez consulter [Create a Deployment Configuration](#).

Fonctions de hook de cycle de vie de déploiement

La section Hooks spécifie les fonctions à exécuter pendant les hooks d'événement de cycle de vie. PreTraffic spécifie la fonction qui s'exécute pendant le hook BeforeAllowTraffic. PostTraffic spécifie la fonction qui s'exécute pendant le hook AfterAllowTraffic.

Autorisations permettant à Lambda d'appeler une autre fonction Lambda

L'lambda:InvokeFunction autorisation spécifiée accorde au rôle utilisé par l'application AWS SAM l'autorisation d'appeler une fonction Lambda. Cela est nécessaire lorsque les CodeDeployHook_afterAllowTraffic fonctions CodeDeployHook_beforeAllowTraffic et invoquent la fonction Lambda déployée lors des tests de validation.

Créez un fichier pour votre fonction Lambda

Créez le fichier pour la fonction que vous mettrez à jour et déployerez ultérieurement dans ce didacticiel.

 Note

Une fonction Lambda peut utiliser n'importe quel environnement d'exécution pris en charge par AWS Lambda. Pour plus d'informations, consultez [Environnement d'exécution AWS Lambda](#).

Pour créer votre fonction Lambda

1. Créez un fichier texte et enregistrez-le sous le nom `myDateTimeFunction.js` dans le répertoire `SAM-Tutorial`.
2. Copiez le code Node.js suivant dans `myDateTimeFunction.js`.

```
'use strict';

exports.handler = function(event, context, callback) {

  if (event.body) {
    event = JSON.parse(event.body);
  }

  var sc; // Status code
  var result = ""; // Response payload

  switch(event.option) {
    case "date":
      switch(event.period) {
        case "yesterday":
          result = setDateResult("yesterday");
          sc = 200;
          break;
        case "today":
          result = setDateResult();
          sc = 200;
          break;
        case "tomorrow":
          result = setDateResult("tomorrow");
          sc = 200;
          break;
        default:
          result = {
            "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
          };
          sc = 400;
          break;
      }
      break;

    /* Later in this tutorial, you update this function by uncommenting
       this section. The framework created by AWS SAM detects the update
       and triggers a deployment by CodeDeploy. The deployment shifts
       production traffic to the updated version of this function.

       case "time":
       var d = new Date();
       var h = d.getHours();
```

```
        var mi = d.getMinutes();
        var s = d.getSeconds();

        result = {
            "hour": h,
            "minute": mi,
            "second": s
        };
        sc = 200;
        break;
    */
    default:
        result = {
            "error": "Must specify 'date' or 'time'."
        };
        sc = 400;
        break;
    }

    const response = {
        statusCode: sc,
        headers: { "Content-type": "application/json" },
        body: JSON.stringify( result )
    };

    callback(null, response);

    function setDateResult(option) {

        var d = new Date(); // Today
        var mo; // Month
        var da; // Day
        var y; // Year

        switch(option) {
            case "yesterday":
                d.setDate(d.getDate() - 1);
                break;
            case "tomorrow":
                d.setDate(d.getDate() + 1);
            default:
                break;
        }
    }
```

```
    mo = d.getMonth() + 1; // Months are zero offset (0-11)
    da = d.getDate();
    y = d.getFullYear();

    result = {
        "month": mo,
        "day": da,
        "year": y
    };

    return result;
}
};
```

La fonction Lambda renvoie le jour, le mois et l'année pour hier, aujourd'hui ou demain. Plus loin dans ce didacticiel, vous annulerez la mise en commentaire du code qui met à jour la fonction pour renvoyer des informations sur le jour ou l'heure que vous spécifiez (par exemple, le jour, le mois et l'année, ou l'heure, la minute et la seconde actuelles). Le framework créé par AWS SAM détecte et déploie la version mise à jour de la fonction.

Note

Cette fonction Lambda est également utilisée dans un AWS Cloud9 didacticiel. AWS Cloud9 est un environnement de développement intégré basé sur le cloud. Pour plus d'informations sur la création, l'exécution, la mise à jour et le débogage de cette fonction dans AWS Cloud9, consultez le [AWS Lambda didacticiel pour AWS Cloud9](#).

Créez un fichier pour votre fonction BeforeAllowTraffic Lambda

Créez le fichier pour la fonction Lambda de votre beforeAllowTraffic hook.

1. Créez un fichier texte et enregistrez-le sous le nom `beforeAllowTraffic.js` dans le répertoire `SAM-Tutorial`.
2. Copiez le code Node.js suivant dans `beforeAllowTraffic.js`. Cette fonction s'exécute pendant le hook BeforeAllowTraffic de votre déploiement.

```
'use strict';
```

```
const AWS = require('aws-sdk');
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
var lambda = new AWS.Lambda();

exports.handler = (event, context, callback) => {

  console.log("Entering PreTraffic Hook!");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event
  payload
  var deploymentId = event.DeploymentId;
  var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

  var functionToTest = process.env.NewVersion;
  console.log("BeforeAllowTraffic hook tests started");
  console.log("Testing new function version: " + functionToTest);

  // Create parameters to pass to the updated Lambda function that
  // include the newly added "time" option. If the function did not
  // update, then the "time" option is invalid and function returns
  // a statusCode of 400 indicating it failed.
  var lambdaParams = {
    FunctionName: functionToTest,
    Payload: "{\"option\": \"time\"}",
    InvocationType: "RequestResponse"
  };

  var lambdaResult = "Failed";
  // Invoke the updated Lambda function.
  lambda.invoke(lambdaParams, function(err, data) {
    if (err){ // an error occurred
      console.log(err, err.stack);
      lambdaResult = "Failed";
    }
    else{ // successful response
      var result = JSON.parse(data.Payload);
      console.log("Result: " + JSON.stringify(result));
      console.log("statusCode: " + result.statusCode);

      // Check if the status code returned by the updated
      // function is 400. If it is, then it failed. If
      // is not, then it succeeded.
      if (result.statusCode != "400"){
        console.log("Validation succeeded");
      }
    }
  });
}
```

```
        lambdaResult = "Succeeded";
    }
    else {
        console.log("Validation failed");
    }

    // Complete the PreTraffic Hook by sending CodeDeploy the validation status
    var params = {
        deploymentId: deploymentId,
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: lambdaResult // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
    if (err) {
        // Validation failed.
        console.log("CodeDeploy Status update failed");
        console.log(err, err.stack);
        callback("CodeDeploy Status update failed");
    } else {
        // Validation succeeded.
        console.log("CodeDeploy status updated successfully");
        callback(null, "CodeDeploy status updated successfully");
    }
    });
}
});
}
```

Créez un fichier pour votre fonction AfterAllowTraffic Lambda

Créez le fichier pour la fonction Lambda de votre afterAllowTraffic hook.

1. Créez un fichier texte et enregistrez-le sous le nom `afterAllowTraffic.js` dans le répertoire `SAM-Tutorial`.
2. Copiez le code Node.js suivant dans `afterAllowTraffic.js`. Cette fonction s'exécute pendant le hook AfterAllowTraffic de votre déploiement.

```
'use strict';
```

```
const AWS = require('aws-sdk');
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
var lambda = new AWS.Lambda();

exports.handler = (event, context, callback) => {

  console.log("Entering PostTraffic Hook!");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event
  // payload
  var deploymentId = event.DeploymentId;
  var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

  var functionToTest = process.env.NewVersion;
  console.log("AfterAllowTraffic hook tests started");
  console.log("Testing new function version: " + functionToTest);

  // Create parameters to pass to the updated Lambda function that
  // include the original "date" parameter. If the function did not
  // update as expected, then the "date" option might be invalid. If
  // the parameter is invalid, the function returns
  // a statusCode of 400 indicating it failed.
  var lambdaParams = {
    FunctionName: functionToTest,
    Payload: "{\"option\": \"date\", \"period\": \"today\"}",
    InvocationType: "RequestResponse"
  };

  var lambdaResult = "Failed";
  // Invoke the updated Lambda function.
  lambda.invoke(lambdaParams, function(err, data) {
    if (err){ // an error occurred
      console.log(err, err.stack);
      lambdaResult = "Failed";
    }
    else{ // successful response
      var result = JSON.parse(data.Payload);
      console.log("Result: " + JSON.stringify(result));
      console.log("statusCode: " + result.statusCode);

      // Check if the status code returned by the updated
      // function is 400. If it is, then it failed. If
      // is not, then it succeeded.
      if (result.statusCode != "400"){
```



```
        console.log("Validation of time parameter succeeded");
        lambdaResult = "Succeeded";
    }
    else {
        console.log("Validation failed");
    }

    // Complete the PostTraffic Hook by sending CodeDeploy the validation status
    var params = {
        deploymentId: deploymentId,
        lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
        status: lambdaResult // status can be 'Succeeded' or 'Failed'
    };

    // Pass CodeDeploy the prepared validation test results.
    codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
    if (err) {
        // Validation failed.
        console.log("CodeDeploy Status update failed");
        console.log(err, err.stack);
        callback("CodeDeploy Status update failed");
    } else {
        // Validation succeeded.
        console.log("CodeDeploy status updated successfully");
        callback(null, "CodeDeploy status updated successfully");
    }
    });
}
});
}
```

Package de l'application AWS SAM

Vous devez maintenant avoir quatre fichiers dans votre répertoire SAM-Tutorial :

- beforeAllowTraffic.js
- afterAllowTraffic.js
- myDateTimeFunction.js
- template.yml

Vous êtes maintenant prêt à utiliser la `sam package` commande AWS SAM pour créer et emballer des artefacts pour vos fonctions et CodeDeploy applications Lambda. Les artefacts sont chargés dans un compartiment S3. La sortie de la commande est un nouveau fichier nommé `package.yml`. Ce fichier est utilisé par la `sam deploy` commande AWS SAM à l'étape suivante.

Note

Pour plus d'informations sur la `sam package` commande, consultez la [référence des commandes AWS SAM CLI](#) dans le manuel du AWS Serverless Application Model développeur.

Dans le répertoire `SAM-Tutorial`, exécutez la commande suivante.

```
sam package \  
  --template-file template.yml \  
  --output-template-file package.yml \  
  --s3-bucket amzn-s3-demo-bucket
```

Pour le `s3-bucket` paramètre, spécifiez le compartiment Amazon S3 que vous avez créé comme condition préalable à ce didacticiel. `output-template-file` Spécifie le nom du nouveau fichier utilisé par la `sam deploy` commande AWS SAM.

Déployer l'application AWS SAM

Utilisez la `sam deploy` commande AWS SAM avec le `package.yml` fichier pour créer vos fonctions Lambda, votre CodeDeploy application et votre groupe de déploiement à l'aide de CloudFormation

Note

Pour plus d'informations sur la `sam deploy` commande, consultez la [référence des commandes AWS SAM CLI](#) dans le manuel du AWS Serverless Application Model développeur.

Dans le répertoire `SAM-Tutorial`, exécutez la commande suivante.

```
sam deploy \  
  --template-file package.yml \  
  --s3-bucket amzn-s3-demo-bucket
```

```
--stack-name my-date-time-app \  
--capabilities CAPABILITY_IAM
```

Le `--capabilities CAPABILITY_IAM` paramètre est obligatoire pour autoriser la création CloudFormation de rôles IAM.

(Facultatif) inspectez et testez votre infrastructure

Cette rubrique explique comment visualiser les composants de votre infrastructure et tester votre fonction Lambda.

Pour afficher le résultat de votre pile après avoir exécuté **sam deploy**

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez Stack (Piles). La pile `my-date-time-app` s'affiche en haut.
3. Choisissez l'onglet Événements pour voir quels événements sont terminés. Vous pouvez afficher les événements pendant que la création de la pile est en cours. Lorsque la création de la pile est terminée, vous pouvez voir tous les événements de création de la pile.
4. Une fois la pile sélectionnée, choisissez Ressources. Dans la colonne Type, vous pouvez voir vos fonctions Lambda, `myDateTimeFunctionCodeDeployHook_beforeAllowTraffic`, et `CodeDeployHook_afterAllowTraffic`. La colonne Physical ID de chacune de vos fonctions Lambda contient un lien permettant d'afficher les fonctions dans la console Lambda.

Note

Le nom de la fonction `myDateTimeFunction` Lambda est précédé du nom de la CloudFormation pile et un identifiant y est ajouté, ce qui donne l'apparence suivante : `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. Ouvrez la CodeDeploy console à l'adresse <https://console.aws.amazon.com/codedeploy/>.
6. Dans le volet de navigation, développez Déployer, puis choisissez Applications.
7. Vous devriez voir une nouvelle CodeDeploy application créée par CloudFormation et dont le nom commence par `my-date-time-app-ServerlessDeploymentApplication`. Choisissez cette application.
8. Vous devez voir un groupe de déploiement dont le nom commence par `my-date-time-app-myDateTimeFunctionDeploymentGroup`. Choisissez ce groupe de déploiement.

Sous Configuration du déploiement, vous devriez voir CodeDeployDefault.LambdaLinear10PercentEvery 1 minute.

(Facultatif) pour tester votre fonction (console)

1. Ouvrez la AWS Lambda console à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le volet de navigation, choisissez votre fonction my-date-time-app-myDateTimeFunction. Dans la console, son nom contient un identifiant. Il ressemble donc à my-date-time-app-myDateTimeFunction-123456ABCDEF.
3. Sélectionnez Tester).
4. Dans Nom de l'événement, entrez un nom pour votre événement de test.
5. Entrez les informations suivantes pour votre événement de test, puis choisissez Créer.

```
{
  "option": "date",
  "period": "today"
}
```

6. Sélectionnez Tester). Vous devriez voir uniquement votre événement de test dans la liste des événements de test.

Pour Résultat de l'exécution, vous devriez voir réussite.

7. Sous Résultat de l'exécution, développez Détails pour afficher les résultats. Vous devriez voir le mois, le jour et l'année en cours.

(Facultatif) pour tester votre fonction (AWS CLI)

1. Localisez l'ARN de votre fonction Lambda. Il apparaît en haut de la console Lambda lorsque vous visualisez votre fonction.
2. Exécutez la commande suivante. Remplacez *your-function-arn* par la fonction ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload '{"option": "date", "period": "today"}' out.txt
```

3. Ouvrez out.txt pour confirmer que le résultat contient le mois, le jour et l'année en cours.

Étape 2 : mise à jour de la fonction Lambda

Dans cette rubrique, vous mettez à jour votre fichier `myDateTimeFunction.js`. Dans l'étape suivante, vous allez utiliser ce fichier pour déployer la fonction mise à jour. Cela déclenche son déploiement CodeDeploy en transférant le trafic de production de la version actuelle de la fonction Lambda vers la version mise à jour.

Pour mettre à jour votre fonction Lambda

1. Ouvrir `myDateTimeFunction.js`.
2. Supprimez les deux marqueurs de commentaire (« `/*` » et « `*/` ») et le texte explicatif au début et à la fin de l'élément `case` nommé `time` dans le bloc `switch`.

Le code sans commentaire vous permet de transmettre un nouveau paramètre, `time`, à la fonction. Si vous transmettez `time` à la fonction mise à jour, elle renvoie les valeurs `hour`, `minute` et `second` actuelles.

3. Enregistrez `myDateTimeFunction.js`. Il devrait se présenter comme suit :

```
'use strict';

exports.handler = function(event, context, callback) {

    if (event.body) {
        event = JSON.parse(event.body);
    }

    var sc; // Status code
    var result = ""; // Response payload

    switch(event.option) {
        case "date":
            switch(event.period) {
                case "yesterday":
                    result = setDateResult("yesterday");
                    sc = 200;
                    break;
                case "today":
                    result = setDateResult();
                    sc = 200;
                    break;
                case "tomorrow":
```

```
        result = setDateResult("tomorrow");
        sc = 200;
        break;
    default:
        result = {
            "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
        };
        sc = 400;
        break;
    }
    break;
    case "time":
        var d = new Date();
        var h = d.getHours();
        var mi = d.getMinutes();
        var s = d.getSeconds();

        result = {
            "hour": h,
            "minute": mi,
            "second": s
        };
        sc = 200;
        break;

    default:
        result = {
            "error": "Must specify 'date' or 'time'."
        };
        sc = 400;
        break;
    }

    const response = {
        statusCode: sc,
        headers: { "Content-type": "application/json" },
        body: JSON.stringify( result )
    };

    callback(null, response);

    function setDateResult(option) {

        var d = new Date(); // Today
```

```
var mo; // Month
var da; // Day
var y; // Year

switch(option) {
  case "yesterday":
    d.setDate(d.getDate() - 1);
    break;
  case "tomorrow":
    d.setDate(d.getDate() + 1);
  default:
    break;
}

mo = d.getMonth() + 1; // Months are zero offset (0-11)
da = d.getDate();
y = d.getFullYear();

result = {
  "month": mo,
  "day": da,
  "year": y
};

return result;
};
```

Étape 3 : Déploiement de la fonction Lambda mise à jour

Au cours de cette étape, vous utiliserez votre mise `myDateTimeFunction.js` à jour pour mettre à jour et lancer le déploiement de votre fonction Lambda. Vous pouvez suivre la progression du déploiement dans la AWS Lambda console CodeDeploy OR.

La `AutoPublishAlias` : `live` ligne de votre AWS SAM modèle permet à votre infrastructure de détecter les mises à jour des fonctions qui utilisent l'`livealias`. Une mise à jour de votre fonction déclenche un déploiement CodeDeploy qui déplace le trafic de production de la version d'origine de la fonction vers la version mise à jour.

Les commandes `sam package` et `sam deploy` sont utilisées pour mettre à jour et déclencher le déploiement de votre fonction Lambda. Vous avez exécuté ces commandes dans [Package de l'application AWS SAM](#) et [Déployer l'application AWS SAM](#).

Pour déployer votre fonction Lambda mise à jour

1. Dans le répertoire `SAM-Tutorial`, exécutez la commande suivante.

```
sam package \  
  --template-file template.yml \  
  --output-template-file package.yml \  
  --s3-bucket amzn-s3-demo-bucket
```

Cela crée un nouvel ensemble d'artefacts qui font référence à votre fonction Lambda mise à jour dans votre compartiment S3.

2. Dans le répertoire `SAM-Tutorial`, exécutez la commande suivante.

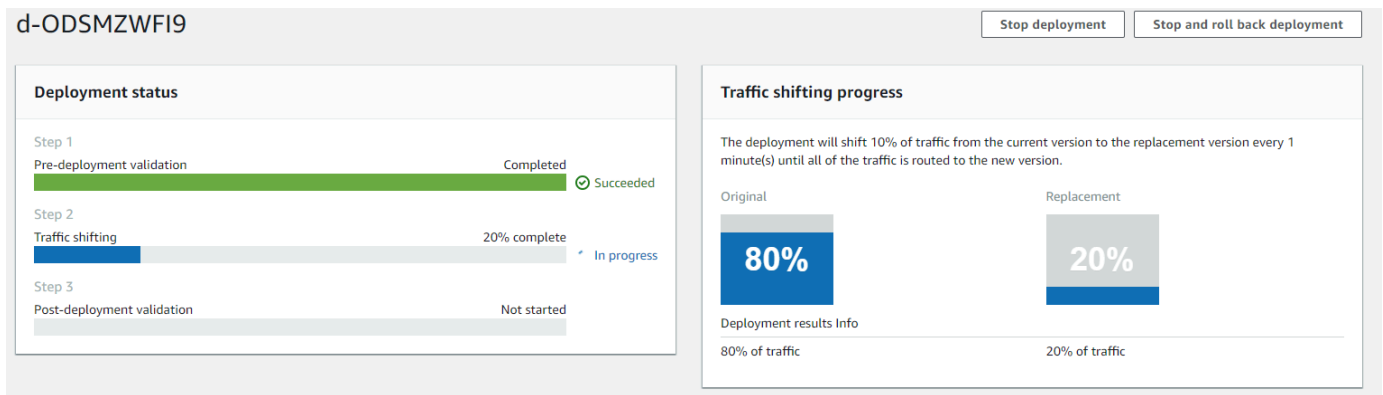
```
sam deploy \  
  --template-file package.yml \  
  --stack-name my-date-time-app \  
  --capabilities CAPABILITY_IAM
```

Comme le nom de la pile est toujours `my-date-time-app`, CloudFormation reconnaît qu'il s'agit d'une mise à jour de la pile. Pour afficher votre pile mise à jour, retournez à la CloudFormation console et, dans le volet de navigation, choisissez **Stacks**.

(Facultatif) pour afficher le trafic lors d'un déploiement (CodeDeploy console)

1. Ouvrez la CodeDeploy console à l'adresse <https://console.aws.amazon.com/codedeploy/>.
2. Dans le volet de navigation, développez **Applications**, puis choisissez votre `my-date-time-app-ServerlessDeploymentApplication` application.
3. Dans **Groupes de déploiement**, choisissez le groupe de déploiement de votre application. Son statut doit être **En cours**.
4. Dans **Deployment group history** (Historique des groupes de déploiement), choisissez le déploiement qui est en cours.

La barre de progression **Traffic shifting** (Déplacement du trafic) et les pourcentages affichés dans les zones **Initiale** et **Remplacement** de cette page indiquent sa progression.



(Facultatif) pour afficher le trafic lors d'un déploiement (console Lambda)

1. Ouvrez la AWS Lambda console à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le volet de navigation, choisissez votre fonction `my-date-time-app-myDateTimeFunction`. Dans la console, son nom contient un identifiant. Il ressemble donc à `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Choisissez Aliases, puis choisissez Live.

Les pondérations en regard de votre version de fonction d'origine (version 1) et de votre version de fonction mise à jour (version 2) indiquent la quantité de trafic transmise à chaque version au moment du chargement de cette page de la console AWS Lambda . La page ne met pas à jour les pondérations au fil du temps. Si vous actualisez la page une fois par minute, la pondération de la version 1 diminue de 10 % et la pondération de la version 2 augmente de 10 % jusqu'à ce que la pondération de la version 2 soit égale à 100.

Aliases

You are viewing the configuration for alias **live**.
[Manage the configuration](#) for the underlying version 1.
[Manage the configuration](#) for the underlying version 2.

Name
live

Description

Version*

1 ▼

Weight: 80%

You can shift traffic between two versions, based on weights (%) that you assign. Click [here](#) to learn more.

Additional version

2 ▼

Weight

20

 %

Étape 4 : Afficher les résultats de votre déploiement

Au cours de cette étape, vous affichez les résultats de votre déploiement. Si votre déploiement réussit, vous pouvez vérifier que votre fonction Lambda mise à jour reçoit le trafic de production. Si votre déploiement échoue, vous pouvez utiliser CloudWatch les journaux pour consulter le résultat des tests de validation de la fonction Lambda exécutés pendant les accrocs du cycle de vie de votre déploiement.

Rubriques

- [Testez votre fonction déployée](#)
- [Afficher les événements liés aux livres dans CloudWatch Logs](#)

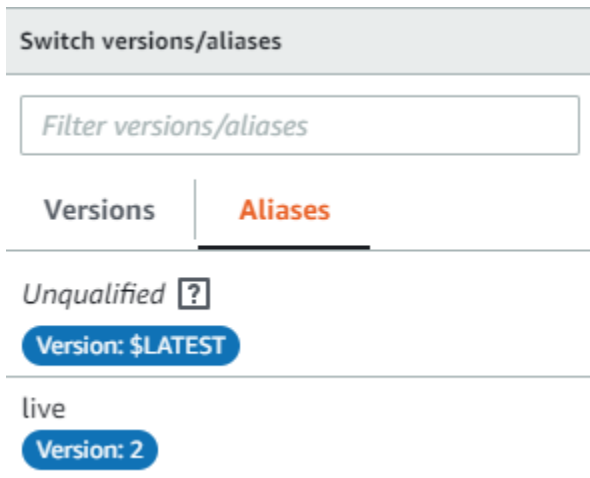
Testez votre fonction déployée

La `aws deploy` commande met à jour la fonction `my-date-time-app-myDateTimeFunction` Lambda. La version de la fonction est mise à jour pour passer à 2 et ajoutée à l'alias `live`.

Pour voir la mise à jour dans la console Lambda

1. Ouvrez la AWS Lambda console à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le volet de navigation, choisissez la fonction `my-date-time-app-myDateTimeFunction`. Dans la console, son nom contient un identifiant. Il ressemble donc à `my-date-time-app-myDateTimeFunction-123456ABCDEF`.

3. Choisissez Qualificateurs, puis Alias. Une fois le déploiement terminé (au bout de 10 minutes environ), pour l'alias `live`, vous devriez voir Version : 2.



4. Dans Code de fonction, affichez le code source de votre fonction. Vos modifications devraient apparaître.
5. (Facultatif) Vous pouvez utiliser les instructions de test de [Étape 2 : mise à jour de la fonction Lambda](#) pour tester votre fonction mise à jour. Créez un nouvel événement de test avec la charge utile suivante, puis confirmez que le résultat contient l'heure, la minute et la seconde actuelles.

```
{
  "option": "time"
}
```

Pour tester la fonction mise AWS CLI à jour, exécutez la commande suivante, puis ouvrez-la out.txt pour confirmer que le résultat contient l'heure, la minute et la seconde actuelles.

```
aws lambda invoke --function your-function-arn --payload '{"option": "time"}'
out.txt
```

Note

Si vous utilisez le AWS CLI pour tester votre fonction avant la fin du déploiement, vous risquez de recevoir des résultats inattendus. Cela est dû au fait que 10 % du trafic passe CodeDeploy progressivement à la version mise à jour chaque minute. Pendant le déploiement, une partie du trafic pointe toujours vers la version d'origine. `aws lambda`

`invoke` peut donc utiliser la version d'origine. Après 10 minutes, le déploiement est terminé et tout le trafic pointe vers la nouvelle version de la fonction.

Afficher les événements liés aux livres dans CloudWatch Logs

Pendant le `BeforeAllowTraffic` hook, CodeDeploy exécute votre fonction `CodeDeployHook_beforeAllowTraffic` Lambda. Pendant le `AfterAllowTraffic` hook, CodeDeploy exécute votre fonction `CodeDeployHook_afterAllowTraffic` Lambda. Chaque fonction exécute un test de validation qui appelle la version mise à jour de votre fonction à l'aide du nouveau paramètre `time`. Si la mise à jour de votre fonction Lambda est réussie, l'option `time` ne provoque pas d'erreur et la validation est réussie. Si la fonction n'a pas été mise à jour, le paramètre non reconnu entraîne une erreur et la validation échoue. Ces tests de validation sont fournis uniquement à des fins de démonstration. Vous écrivez vos propres tests pour valider votre déploiement. Vous pouvez utiliser la console CloudWatch Logs pour consulter vos tests de validation.

Pour consulter les événements liés à vos CodeDeploy accrochages

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Logs (Journaux).
3. Dans la liste des groupes de journaux, choisissez `/aws/lambda/CodeDeployHook_beforeAllowTraffic` ou `/aws/lambda/CodeDeployHook_afterAllowTraffic`.
4. Choisissez le flux de journaux. Vous ne devriez en voir qu'un seul.
5. Développez les événements pour afficher leurs détails.

	Time (UTC +00:00)	Message
	2019-07-12	
		No older events found at the moment. Re...
▶	22:08:56	START RequestId: 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Version: \$LATEST
▶	22:08:56	2019-07-12T22:08:56.834Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Entering PreTraffic Hook!
▶	22:08:56	2019-07-12T22:08:56.834Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Testing new function version: arn:aws:lambda:ca-
▼	22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Result: {"statusCode":200,"headers":{"Content-t...
	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Result:	
		{ "statusCode": 200, "headers": { "Content-type": "application/json" }, "body": "{ \"hour\":22, \"minute\":8, \"second\":57 }" }
▼	22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 statusCode: 200
	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49	statusCode: 200
▼	22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Validation succeeded
	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49	Validation succeeded
▼	22:08:58	2019-07-12T22:08:58.302Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Codedeploy status updated successfully
	2019-07-12T22:08:58.302Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49	Codedeploy status updated successfully

Étape 5 : nettoyer

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 les CloudWatch journaux créés par vos fonctions de validation Lambda.

Pour supprimer votre CloudFormation pile

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la colonne Piles, choisissez votre pile my-date-time-app, puis choisissez Supprimer.
3. À l'invite, choisissez Supprimer la pile. Les fonctions Lambda, CodeDeploy l'application et le groupe de déploiement, ainsi que les rôles IAM créés par AWS SAM sont supprimés.

Pour supprimer vos journaux dans CloudWatch Logs

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Logs (Journaux).

3. Dans la liste des groupes de journaux, cliquez sur le bouton situé à côté de `/aws/lambda/CodeDeployHook_beforeAllowTraffic`.
4. Dans Actions, choisissez Supprimer le groupe de journaux, puis choisissez Oui, supprimer.
5. Dans la liste des groupes de journaux, cliquez sur le bouton situé à côté de `/aws/lambda/CodeDeployHook_afterAllowTraffic`.
6. Dans Actions, choisissez Supprimer le groupe de journaux, puis choisissez Oui, supprimer.

Travailler avec l' CodeDeploy agent

L' AWS CodeDeploy agent est un progiciel qui, une fois installé et configuré sur une instance, permet à cette instance d'être utilisée dans CodeDeploy des déploiements.

AWS prend en charge la dernière version mineure de l' CodeDeploy agent. Actuellement, la dernière version mineure est 1.7.x.

Note

L' CodeDeploy agent n'est requis que si vous déployez sur une plate-forme informatique EC2/sur site. L'agent n'est pas requis pour les déploiements utilisant la plateforme de calcul Amazon ECS ou AWS Lambda.

Un fichier de configuration est placé sur l'instance lorsque l'agent &ACD; est installé. Ce fichier est utilisé pour spécifier la manière dont l'agent fonctionne. Ce fichier de configuration spécifie les chemins de répertoire et les autres paramètres AWS CodeDeploy à utiliser lorsqu'il interagit avec l'instance. Vous pouvez modifier certaines options de configuration dans ce fichier. Pour plus d'informations sur l'utilisation du fichier de configuration de l' CodeDeploy agent, consultez [CodeDeploy référence de configuration de l'agent](#).

Pour plus d'informations sur l'utilisation de l' CodeDeploy agent, telles que les étapes d'installation, de mise à jour et de vérification des versions, consultez [Gestion des opérations des CodeDeploy agents](#).

Rubriques

- [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#)
- [Protocole de communication et port pour l' CodeDeploy agent](#)
- [Historique des versions de l' CodeDeploy agent](#)
- [Gérer le CodeDeploy processus](#)
- [Révision de l'application et nettoyage des fichiers journaux](#)
- [Fichiers installés par l' CodeDeploy agent](#)
- [Gestion des opérations des CodeDeploy agents](#)

Systèmes d'exploitation pris en charge par l' CodeDeploy agent

Systèmes d'exploitation Amazon EC2 AMI pris en charge

L' CodeDeploy agent a été testé sur les systèmes d'exploitation Amazon EC2 AMI suivants :

- Amazon Linux 2023 (ARM, x86)
- Amazon Linux 2 (ARM, x86)
- Microsoft Windows Server 2022, 2019
- Red Hat Enterprise Linux (RHEL) 9.x, 8.x, 7.x
- Serveur Ubuntu 22,04 LTS, 20,04 LTS, 18,04 LTS, 16,04 LTS

L' CodeDeploy agent est disponible en open source pour que vous puissiez l'adapter à vos besoins. Il peut être utilisé avec d'autres systèmes d'exploitation AMI Amazon EC2. Pour plus d'informations, accédez au référentiel des [CodeDeploy agents](#) dans GitHub.

Systèmes d'exploitation sur site pris en charge

L' CodeDeploy agent a été testé sur les systèmes d'exploitation locaux suivants :

- Microsoft Windows Server 2022, 2019
- Red Hat Enterprise Linux (RHEL) 9.x, 8.x, 7.x
- Serveur Ubuntu 22.04 LTS, 20.04 LTS

L' CodeDeploy agent est disponible en open source pour que vous puissiez l'adapter à vos besoins. Il peut être utilisé avec d'autres systèmes d'exploitation d'instance sur site. Pour plus d'informations, accédez au référentiel des [CodeDeploy agents](#) dans GitHub.

Protocole de communication et port pour l' CodeDeploy agent

L' CodeDeploy agent communique avec le trafic sortant via le protocole HTTPS via le port 443.

Lorsque l' CodeDeploy agent s'exécute sur une instance EC2, il utilise le point de terminaison des [métadonnées EC2](#) pour récupérer les informations relatives à l'instance. Découvrez-en plus sur la [limitation et l'octroi de l'accès au service des métadonnées d'instance](#).

Historique des versions de l' CodeDeploy agent

Vos instances doivent exécuter une version prise en charge de l' CodeDeploy agent. La version minimale actuellement prise en charge est 1.7.x.

Note

Nous vous recommandons d'utiliser la dernière version de l' CodeDeploy agent. Si vous rencontrez des problèmes, passez à la dernière version avant de contacter le AWS Support. Pour plus d'informations sur la mise à niveau, consultez [Mettre à jour l' CodeDeploy agent](#).

Le tableau suivant répertorie toutes les versions de l' CodeDeploy agent ainsi que les fonctionnalités et améliorations incluses dans chaque version.

Version	Date de publication	Détails
1.8.0	31 juillet 2025	Modifié : mise à niveau du Ruby intégré vers la version 3.2 dans l' CodeDeploy agent pour Windows.
1.7.1	14 novembre 2024	Modifié : dépendances mises à jour pour les correctifs de sécurité.
1.7.0	6 mars 2024	Ajouté : un paramètre <code>:disable_imds_v1:</code> de configuration dans le fichier de configuration de l' CodeDeploy agent. Utilisez ce paramètre pour désactiver la fonction de secours en IMDSv1 cas d' IMDSv2 erreur. La valeur par défaut est <code>false</code> (active le repli). Pour plus d'informations, consultez la référence de configuration de CodeDeploy l'agent. Ajouté : Support pour le système d'exploitation Red Hat Enterprise Linux 9 (RHEL 9). Ajouté : Support pour les versions 3.1 et 3.2 de Ruby sur Ubuntu Server. Corrigé : L' CodeDeploy agent génère désormais une erreur conviviale si le fichier de configuration de l' CodeDeploy agent ne parvient pas à se charger.

Version	Date de publication	Détails
		Modifié : mise à niveau de Ruby vers la version 2.7.8-1 dans l'CodeDeploy agent pour Windows.
1.6.0	30 mars 2023	Ajouté : Support pour Ruby 3.1, 3.2. Ajouté : Support pour Amazon Linux 2023. Ajouté : Support pour Windows Server 2022. Modifié : le paramètre par défaut de <code>verbose</code> est désormais <code>false</code> pour les instances Windows Server. Pour continuer à imprimer les messages de débogage dans les fichiers journaux sous Windows, vous devez <code>verbose</code> définir <code>true</code> sur. Supprimé : Support pour Windows Server 2016 et Windows Server 2012 R2. Supprimé : Support pour Amazon Linux 2018.03.x.

Version	Date de publication	Détails
1.5.0	3 mars 2023	Ajouté : Support pour Ruby 3. Ajouté : Support pour Ubuntu 22.04. Correction : le redémarrage de l' CodeDeploy agent peu après le démarrage entraînait son blocage. Modifié : L' CodeDeploy agent échoue désormais au déploiement d'un hôte au démarrage de l'agent si le service de l'agent redémarre de façon inattendue lors de l'exécution d'un script hook. Ce correctif vous permet d'éviter d'attendre le délai d'expiration de 70 minutes avant de réessayer un déploiement. Avis d'obsolescence : CodeDeploy l'agent 1.5.0 est la dernière version compatible avec Windows Server 2016 et Windows Server 2012 R2. Supprimé : Support de l' CodeDeployagent sur Ubuntu 14.04 LTS, Windows Server 2008 R2 et Windows Server 2008 R2 32 bits.
1.4.1	6 décembre 2022	Corrigé : vulnérabilité de sécurité liée à la journalisation. Amélioration : amélioration de la journalisation lors de l'interrogation de la commande hôte.

Version	Date de publication	Détails
1.4.0	31 août 2022	Ajouté : Support pour Red Hat Enterprise Linux 8. Ajouté : Support pour les longs chemins de fichiers sur l' CodeDeploy agent pour Windows. Pour activer les longs chemins de fichier, vous devez définir la clé de registre Windows appropriée, puis redémarrer votre agent. Pour de plus amples informations, veuillez consulter Les longs chemins de fichier provoquent des erreurs « Aucun fichier ou répertoire de ce type ». Correction : problème lié à l'opération de décompression lorsque le disque était plein. L' CodeDeploy agent détecte désormais le code de sortie 50 du dézip indiquant un disque plein, supprime les fichiers partiellement extraits et déclenche une exception pour signaler une défaillance au CodeDeploy serveur. Le message d'erreur est visible sous la forme d'un message d'erreur lié à un événement du cycle de vie, et le déploiement au niveau de l'hôte s'arrête sans être bloqué ni expiration du délai imparti. Corrigé : un problème qui provoquait l'échec de l'agent. Corrigé : problème à cause duquel les hameçons expiraient en cas de course difficile. Les hooks sans script se poursuivent désormais et ne provoqueront plus d'échecs ni de délais d'attente. Modifié : le update script du bin répertoire de l' CodeDeploy agent a été supprimé car il n'est plus utilisé. Modifié : L' CodeDeploy agent pour Windows Server intègre désormais Ruby 2.7. Modifié : De nouvelles variables d'environnement ont été ajoutées, destinées à être utilisées par les scripts hook en fonction de la source du bundle de déploiement (Amazon S3 ou GitHub).

Version	Date de publication	Détails
		Pour de plus amples informations, veuillez consulter Disponibilité des variables d'environnement pour les hooks.  Important Avis d'obsolescence : l' CodeDeployagent 1.4.0 est la dernière version qui inclura des programmes d'installation pour Windows Server 32 bits. Avis d'obsolescence : CodeDeploy l'agent 1.4.0 est la dernière version compatible avec Windows Server 2008 R2. Supprimé : Support de l' CodeDeploy agent sur les Amazon EC2 suivants : AMIs Amazon Linux 2014.09, 2016.03, 2016.09 et 2017.03.

Version	Date de publication	Détails
1.3.2	6 mai 2021	 Important CodeDeploy l'agent 1.3.2 résout le problème CVE-2018-1000201 qui affecte les hôtes Windows exécutant l'agent. Le CVE cite ruby-ffi, qui est une dépendance de l'agent. CodeDeploy Si votre agent a été installé avec Amazon EC2 Systems Manager (SSM) et qu'il est configuré pour être mis à jour automatiquement, aucune action n'est requise. Dans le cas contraire, une action est requise pour mettre à jour manuellement l'agent. Pour mettre à niveau l'agent, suivez les instructions de la section Mettre à jour l' CodeDeploy agent sous Windows Server. Correction : problème lors de l'installation de l' CodeDeploy agent sur Ubuntu 20.04 et versions ultérieures. Corrigé : un problème intermittent survenait lors de l'extraction de fichiers compressés car les chemins relatifs n'étaient pas gérés correctement. Ajouté : Support AWS PrivateLink et points de terminaison VPC pour les instances Windows. Ajouté : améliorations des AppSpec fichiers, comme décrit ci-dessous. • Vous pouvez désormais spécifier un nom de fichier personnalisé pour le AppSpec fichier lors de la création d'un déploiement local. Pour de plus amples informations, veuillez consulter Création d'un déploiement local. • Le AppSpec fichier peut désormais avoir une extension de .yaml fichier.

Version	Date de publication	Détails
		Vous pouvez désormais remplacer les fichiers déployés à l'aide d'un nouveau <code>file_exists_behavior</code> paramètre facultatif dans le AppSpec fichier. Pour de plus amples informations, veuillez consulter AppSpec section « fichiers » (déploiements EC2/sur site uniquement). Mise à niveau : utilise CodeDeploy désormais le AWS SDK pour Ruby 3.0.
1.3.1	22 décembre 2020	Correction : problème 1.3.0 qui empêchait le démarrage des instances locales.
1.3.0	10 novembre 2020	 Important Cette version est obsolète. Corrigé : suppression d'un certificat expiré qui n'était plus utilisé. Corrigé : suppression du message d'invite du script de désinstallation de l'agent utilisé par AWS Systems Manager, ce qui facilite la rétrogradation d'un hôte ou d'un parc vers une version précédente de l'agent.

Version	Date de publication	Détails
1.2.1	23 septembre 2020	Modifié : AWS SDK pour Ruby dépendance améliorée de la v2 à la v3. Ajouté : Support pour IMDSv2. Inclut une solution de secours silencieuse en IMDSv1 cas d'échec des requêtes IMDSv2 HTTP. Modifié : mise à jour des dépendances Rake et Rubzip pour les correctifs de sécurité. Corrigé : assurez-vous qu'un fichier PID vide renvoie l'état de No CodeDeploy Agent Running et nettoie le fichier PID au démarrage de l'agent.

Version	Date de publication	Détails
1.1.2	4 août 2020	Ajouté : Support pour Ubuntu Server 19.10 et 20.04. Remarque : La version 19.10 a atteint sa end-of-life date limite et n'est plus prise en charge par Ubuntu ou CodeDeploy. Ajouté : améliorations de l'efficacité de la mémoire pour Linux et Ubuntu afin de libérer la mémoire réservée plus rapidement. Ajouté : compatibilité avec le « nettoyage silencieux » de Windows Server, qui empêchait l'agent de répondre dans certains cas. Ajouté : Ignorez les répertoires non vides pendant le nettoyage pour éviter les échecs lors du déploiement. Ajouté : Support pour la zone AWS locale de Los Angeles (LA). Ajouté : Extrayez l'AZ des métadonnées de l'instance pour assurer la compatibilité avec les Zones AWS Locales. Ajouté : Les utilisateurs peuvent désormais fournir leurs archives dans des sous-répertoires et ne sont pas tenus de les stocker dans le répertoire racine. Ajouté : Un problème avec Rubyzip susceptible d'entraîner des fuites de mémoire a été détecté. Mise à jour de la commande unzip pour essayer d'abord d'utiliser un utilitaire de décompression installé sur le système avant d'utiliser Rubyzip. Ajouté : <code>:enable_auth_policy:</code> en tant que paramètre de configuration de l'agent. Modifié : les avertissements de décompression sont désormais ignorés, les déploiements se poursuivront donc.

Version	Date de publication	Détails
1.1.0	30 juin 2020	Modifié : la gestion des versions de l' CodeDeploy agent suit désormais la convention de version standard de Ruby. Ajouté : nouveau paramètre de commande d'installation et de mise à jour pour permettre l'installation d'une la version spécifique de l'agent à partir de la ligne de commande. Supprimé : suppression de l' CodeDeploy agent Auto Updater pour Linux et Ubuntu. Pour configurer les mises à jour automatiques de l' CodeDeploy agent, voir Installer l' CodeDeploy agent à l'aide de AWS Systems Manager.
1.0.1.1597	15 novembre 2018	Amélioration : CodeDeploy supporte Ubuntu 18.04. Amélioration : CodeDeploy supporte Ruby 2.5. Amélioration : CodeDeploy prend en charge les points de terminaison FIPS. Pour de plus amples informations sur les points de terminaison FIPS, veuillez consulter Présentation de FIPS 140-2. Pour les points de terminaison pouvant être utilisés avec CodeBuild, voir CodeDeploy Régions et points de terminaison.
1.0.1.1518	12 juin 2018	Amélioration : correction d'un problème qui provoquait une erreur lorsque l' CodeDeploy agent était fermé alors qu'il acceptait des demandes de sondage. Amélioration : Ajout d'une fonctionnalité de suivi du déploiement qui empêche la fermeture de l' CodeDeploy agent lorsqu'un déploiement est en cours. Amélioration : Amélioration des performances lors de la suppression des fichiers.

Version	Date de publication	Détails
1.0.1.1458	6 mars 2018	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : Amélioration des validations de certificat pour prendre en charge un nombre plus important d'autorités approuvées. Amélioration : correction d'un problème qui provoquait l'échec de la CLI locale lors d'un déploiement incluant un événement BeforeInstall du cycle de vie. Amélioration : correction d'un problème qui pouvait entraîner l'échec d'un déploiement actif lors de la mise à jour de l' CodeDeploy agent.
1.0.1.1352	16 novembre 2017	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : introduction d'une nouvelle fonctionnalité permettant de tester et de déboguer un déploiement EC2/ sur site sur une machine ou une instance locale sur laquelle l' CodeDeploy agent est installé.
1.0.1.1106	16 mai 2017	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : Introduction de la nouvelle prise en charge de la gestion des contenus dans un emplacement cible qui ne faisait pas partie de la révision de l'application depuis le déploiement réussi le plus récent. Les options de déploiement pour le contenu existant incluent désormais la conservation du contenu, le remplacement du contenu ou l'échec du déploiement. Amélioration : rendu l' CodeDeploy agent compatible avec la version 2.9.2 du AWS SDK pour Ruby (aws-sdk-core2.9.2).

Version	Date de publication	Détails
1.0.1.1095	29 mars 2017	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : introduction du support pour l' CodeDeploy agent dans la région de Chine (Pékin). Amélioration : Puppet a été activé pour s'exécuter sur des instances Windows Server lorsqu'il est invoqué par un hook d'événements du cycle de vie. Amélioration: Amélioration de la gestion des <code>untar</code> opérations.
1.0.1.1067	6 janvier 2017	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : révision de nombreux messages d'erreur afin de décrire plus précisément l'origine de l'échec du déploiement. Amélioration : correction d'un problème qui empêchait l' CodeDeploy agent d'identifier la révision d'application correcte à déployer lors de certains déploiements. Amélioration : rétablissement de l'utilisation de <code>pushd</code> et <code>popd</code> avant et après l'opération <code>untar</code>.
1.0.1.1045	21 novembre 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : rendu l' CodeDeploy agent compatible avec la version 2.6.11 du AWS SDK pour Ruby (<code>aws-sdk-core2.6.11</code>).

Version	Date de publication	Détails
1.0.1.1037	19 octobre 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. L' CodeDeploy agent pour les instances Amazon Linux, RHEL et Ubuntu Server a été mis à jour avec les modifications suivantes. Pour les instances Windows Server, la dernière version reste 1.0.1.998. Amélioration : l'agent peut désormais déterminer quelle version de Ruby est installée sur une instance afin de s'en servir pour appeler le script <code>codedeploy-agent</code> .
1.0.1.101 1.1	17 août 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : suppression des modifications apportées par la version 1.0.1.1011 en raison de problèmes liés à la prise en charge du shell. Cette version de l'agent équivaut d'un point de vue fonctionnel à la version 1.0.1.998 publiée le 11 juillet 2016.

Version	Date de publication	Détails
1.0.1.1011	15 août 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. L' CodeDeploy agent pour les instances Amazon Linux, RHEL et Ubuntu Server a été mis à jour avec les modifications suivantes. Pour les instances Windows Server, la dernière version reste 1.0.1.998. Fonctionnalité : Ajout de la prise en charge de l'appel de l' CodeDeploy agent à l'aide du shell bash sur les systèmes d'exploitation sur lesquels le système d'initialisation systemd est utilisé. Amélioration : prise en charge de toutes les versions de Ruby 2.x dans l' CodeDeploy agent et dans le programme de mise à jour de l' CodeDeploy agent. Les CodeDeploy agents mis à jour ne dépendent plus uniquement de Ruby 2.0. (Ruby 2.0 est toujours requis pour les versions deb et rpm du programme d'installation de l' CodeDeploy agent.)
1.0.1.998	11 juillet 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Amélioration : correction de la prise en charge de l'exécution de l' CodeDeploy agent avec des profils utilisateur autres que root. La variable nommée USER est remplacée par CODEDEPLOY_USER pour éviter les conflits avec les variables environnementales.

Version	Date de publication	Détails
1.0.1.966	16 juin 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : Ajout de la prise en charge de l'exécution de l'CodeDeploy agent avec des profils utilisateur autres que root. Amélioration : Correction de la prise en charge pour spécifier le nombre de révisions d'application que vous voulez que l'agent CodeDeploy archive pour un groupe de déploiement. Amélioration : rendu l' CodeDeploy agent compatible avec la version 2.3 du AWS SDK pour Ruby (aws-sdk-core 2.3). Amélioration : Correction des problèmes liés à l'encodage UTF-8 au cours des déploiements. Amélioration : meilleure précision lors de l'identification des noms de processus.
1.0.1.950	24 mars 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : Ajout de la prise en charge du serveur proxy d'installation. Amélioration : le script d'installation a été mis à jour afin de ne pas télécharger l' CodeDeploy agent si la dernière version est déjà installée.
1.0.1.934	11 février 2016	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : Introduction de la prise en charge pour spécifier le nombre de révisions d'application que vous voulez que l'agent CodeDeploy archive pour un groupe de déploiement.

Version	Date de publication	Détails
1.0.1.880	11 janvier 2016	Remarque : Cette version n'est plus prise en charge et peut entraîner l'échec des déploiements. Amélioration : rendu l' CodeDeploy agent compatible avec la version 2.2 du AWS SDK pour Ruby (aws-sdk-core 2.2). La version 2.1.2 est encore prise en charge.
1.0.1.854	17 novembre 2015	Remarque : Cette version n'est plus prise en charge. Si vous utilisez cette version, vos déploiements peuvent échouer. Fonctionnalité : Introduction de la prise en charge pour l'algorithme de hachage SHA-256. Fonctionnalité : Introduction de la prise en charge du suivi des versions dans les fichiers <code>.version</code>. Fonctionnalité : Mise à disposition de l'ID de groupe de déploiement via l'utilisation d'une variable d'environnement. Amélioration : Ajout de la prise en charge de la surveillance des journaux des CodeDeploy agents à l'aide d'Amazon CloudWatch Logs.

Pour obtenir des informations connexes, consultez les ressources suivantes :

- [Déterminer la version de l' CodeDeploy agent](#)
- [Installation de l' CodeDeploy agent](#)

Pour un historique des versions de l' CodeDeploy agent, consultez le [référentiel des versions sur GitHub](#).

Gérer le CodeDeploy processus

Toutes les distributions Linux de l' CodeDeploy agent (rpm et deb) utilisent [systemd](#) par défaut pour gérer le processus de l'agent.

Cependant, les distributions rpm et deb sont fournies avec des scripts de démarrage résidant à `/etc/init.d/codedeploy-agent`. Selon la distribution que vous utilisez, lorsque vous utilisez une commande telle que `sudo service codedeploy-agent restart`, les scripts `/etc/init.d` peuvent être exécutés pour lancer le processus de l'agent au lieu d'autoriser la gestion du processus. `/etc/init.d` n'est pas souhaitable d'exécuter des scripts sur.

Pour éviter ce problème, pour les systèmes compatibles, `systemd` nous recommandons d'utiliser l'utilitaire `systemctl` pour toutes les opérations de l'agent au lieu d'utiliser la commande `service`.

Par exemple, pour redémarrer l'agent CodeDeploy, utilisez `sudo systemctl restart codedeploy-agent` au lieu de la commande équivalente avec l'utilitaire `service`.

Révision de l'application et nettoyage des fichiers journaux

L'agent CodeDeploy archive les révisions et les fichiers journaux sur les instances. L'agent CodeDeploy nettoie ces artefacts pour économiser de l'espace disque.

Journaux de déploiement des révisions d'applications : vous pouvez utiliser l'option `max_revisions` : dans le fichier de configuration de l'agent pour spécifier le nombre de révisions d'applications à archiver en saisissant un entier positif. CodeDeploy archive également les fichiers journaux de ces révisions. Tous les autres sont supprimés, à l'exception du fichier journal du dernier déploiement réussi. Ce fichier journal est toujours conservé, même si le nombre de déploiements ayant échoué dépasse le nombre de révisions conservées. Si aucune valeur n'est spécifiée, CodeDeploy conserve les cinq révisions les plus récentes en plus de la révision actuellement déployée.

CodeDeploy journaux : pour les instances Amazon Linux, Ubuntu Server et RHEL, l'agent CodeDeploy fait pivoter les fichiers journaux sous le dossier `/var/log/aws/codedeploy-agent`. Le fichier journal est permuté tous les jours à 00:00:00 (heure de l'instance). Les fichiers journaux sont supprimés après sept jours. Le modèle d'affectation de noms des fichiers journaux permutés est `codedeploy-agent.YYYYMMDD.log`.

Fichiers installés par l'agent CodeDeploy

L'agent CodeDeploy stocke les révisions, l'historique des déploiements et les scripts de déploiement dans son répertoire racine sur une instance. Le nom et l'emplacement par défaut de cet annuaire sont :

`'/opt/codedeploy-agent/deployment-root'` pour les instances Amazon Linux, Ubuntu Server et RHEL.

'C:\ProgramData\Amazon\CodeDeploy' pour les instances de Windows Server.

Vous pouvez utiliser le paramètre `root_dir` dans le fichier de configuration de l'agent CodeDeploy pour configurer le nom et l'emplacement du répertoire. Pour de plus amples informations, veuillez consulter [CodeDeploy référence de configuration de l'agent](#).

Ci-dessous figure un exemple de la structure du fichier et de l'annuaire sous l'annuaire racine. La structure suppose qu'il existe un nombre N de groupes de déploiement et que chaque groupe de déploiement comprend N déploiements.

```
|--deployment-root/
|-- deployment group 1 ID
|   |-- deployment 1 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment 2 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment N ID
|   |   |-- Contents and logs of the deployment's revision
|-- deployment group 2 ID
|   |-- deployment 1 ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   |-- contents of the deployment's revision
|   |   |-- logs
|   |   |   |-- scripts.log
|   |-- deployment 2 ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   |-- contents of the deployment's revision
|   |   |-- logs
|   |   |   |-- scripts.log
|   |-- deployment N ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   |-- contents of the deployment's revision
|   |   |-- logs
|   |   |   |-- scripts.log
|-- deployment group N ID
|   |-- deployment 1 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment 2 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment N ID
```

```
|      |      |-- Contents and logs of the deployment's revision
|-- deployment-instructions
|      |-- [deployment group 1 ID]_cleanup
|      |-- [deployment group 2 ID]_cleanup
|      |-- [deployment group N ID]_cleanup
|      |-- [deployment group 1 ID]_install.json
|      |-- [deployment group 2 ID]_install.json
|      |-- [deployment group N ID]_install.json
|      |-- [deployment group 1 ID]_last_successful_install
|      |-- [deployment group 2 ID]_last_successful_install
|      |-- [deployment group N ID]_last_successful_install
|      |-- [deployment group 1 ID]_most_recent_install
|      |-- [deployment group 2 ID]_most_recent_install
|      |-- [deployment group N ID]_most_recent_install
|-- deployment-logs
|      |-- codedeploy-agent-deployments.log
```

- Les dossiers relatifs aux ID de groupes de déploiement représentent chacun de vos groupes de déploiement. Le nom d'un annuaire de groupes de déploiement est son ID (par exemple, `acde1916-9099-7caf-fd21-012345abcdef`). Un annuaire de groupe de déploiement abrite un sous-annuaire pour chaque tentative de déploiement dans ce groupe.

Vous pouvez utiliser la commande [batch-get-deployments](#) pour rechercher un ID de groupe de déploiement.

- Les dossiers relatifs aux ID de déploiement représentent chacun des déploiements d'un groupe. Le nom d'un annuaire de déploiement est son ID. Chaque dossier contient les éléments suivants :
 - `bundle.tar`, un fichier compressé hébergeant le contenu de la révision de déploiement. Utilisez un utilitaire de décompression zip si vous souhaitez afficher la révision.
 - `deployment-archive`, un annuaire comportant le contenu de la révision de déploiement.
 - `logs`, un annuaire contenant un fichier `scripts.log`. Ce fichier répertorie les résultats de tous les scripts spécifiés dans le AppSpec fichier de déploiement.

Si vous souhaitez rechercher le dossier d'un déploiement mais que vous ne connaissez pas son ID de déploiement ou son ID de groupe de déploiement, vous pouvez utiliser la [AWS CodeDeploy console](#) ou le AWS CLI pour les trouver. Pour de plus amples informations, veuillez consulter [Afficher les détails CodeDeploy du déploiement](#).

Le nombre maximal par défaut de déploiements pouvant être archivés dans un groupe de déploiements est cinq. Lorsque ce nombre est atteint, les déploiements ultérieurs écrasent l'archive.

la plus ancienne. Vous pouvez utiliser le paramètre `max_revisions` dans le fichier de configuration de l' `CodeDeploy` agent pour modifier la valeur par défaut. Pour de plus amples informations, veuillez consulter [CodeDeploy référence de configuration de l'agent](#).

Note

Si vous souhaitez récupérer de l'espace de disque dur utilisé par les déploiements archivés, mettez à jour le paramètre `max_revisions` sur une valeur plus faible, comme 1 ou 2. Le prochain déploiement écrasant les déploiements archivés, le nombre est égal à la valeur spécifiée.

- `deployment-instructions` comporte quatre fichiers texte pour chacun des groupes de déploiement :
 - `[Deployment Group ID]-cleanup`, un fichier texte avec une version d'annulation de chaque commande exécutée durant un déploiement. Comme nom de fichier, citons par exemple `acde1916-9099-7caf-fd21-012345abcdef-cleanup`.
 - `[Deployment Group ID]-install.json`, un fichier JSON créé durant le déploiement le plus récent. Il contient les commandes exécutées durant le déploiement. Comme nom de fichier, citons par exemple `acde1916-9099-7caf-fd21-012345abcdef-install.json`.
 - `[Deployment Group ID]_last_successfull_install`, un fichier texte répertoriant l'annuaire d'archivage du dernier déploiement réussi. Ce fichier est créé lorsque l' `CodeDeploy` agent a copié tous les fichiers de l'application de déploiement sur l'instance. Il est utilisé par l' `CodeDeploy` agent lors du prochain déploiement pour déterminer les `BeforeInstall` scripts `ApplicationStop` et les scripts à exécuter. Comme nom de fichier, citons par exemple `acde1916-9099-7caf-fd21-012345abcdef_last_successfull_install`.
 - `[Deployment Group ID]_most_recent_install`, un fichier texte qui répertorie le nom de l'annuaire d'archivage du déploiement le plus récent. Ce fichier est créé une fois que les fichiers du déploiement ont été téléchargés. Le fichier `[deployment group ID]_last_successfull_install` file est créé après ce fichier, une fois que les fichiers téléchargés ont été copiés vers leur destination finale. Comme nom de fichier, citons par exemple `acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.
- `deployment-logs` comporte les fichiers journaux suivants :
 - Les fichiers `codedeploy-agent.yyyymmdd.log` sont créés pour chacun des jours au cours desquels un déploiement est exécuté. Chaque fichier journal comporte des informations sur les déploiements de la journée. Ces fichiers peuvent s'avérer utiles pour le débogage de problèmes, liés par exemple aux autorisations. Le fichier journal est initialement nommé `codedeploy-agent.log`. Le jour suivant, la date de ses déploiements est insérée dans le nom du fichier.

Par exemple, si nous sommes aujourd'hui le 3 janvier 2018, les données sur l'ensemble des déploiements de la journée sont disponibles dans `codedeploy-agent.log`. Demain, le 4 janvier 2018, le fichier journal est renommé `codedeploy-agent.20180103.log`.

- `codedeploy-agent-deployments.log` compile le contenu des `scripts.log` fichiers pour chaque déploiement. Les fichiers `scripts.log` sont hébergés dans le sous-dossier `logs` de chaque dossier `Deployment ID`. Les entrées de ce fichier sont précédées d'un ID de déploiement. Par exemple, « `[d-ABCDEF123]LifecycleEvent - BeforeInstall` » peut être écrit durant le déploiement avec un ID de `d-ABCDEF123`. Lorsque `codedeploy-agent-deployments.log` sa taille maximale est atteinte, l'CodeDeploy agent continue d'y écrire tout en supprimant l'ancien contenu.

Gestion des opérations des CodeDeploy agents

Les instructions de cette section vous indiquent comment installer, désinstaller, réinstaller ou mettre à jour l'CodeDeploy agent et comment vérifier que l'CodeDeploy agent est en cours d'exécution.

Rubriques

- [Vérifiez que l'CodeDeploy agent est en cours d'exécution](#)
- [Déterminer la version de l'CodeDeploy agent](#)
- [Installation de l'CodeDeploy agent](#)
- [Mettre à jour l'CodeDeploy agent](#)
- [Désinstallez l'CodeDeploy agent](#)
- [Envoyer les journaux des CodeDeploy agents à CloudWatch](#)

Vérifiez que l'CodeDeploy agent est en cours d'exécution

Cette section décrit les commandes à exécuter si vous pensez que l'CodeDeploy agent a cessé de s'exécuter sur une instance.

Rubriques

- [Vérifiez que l'CodeDeploy agent pour Amazon Linux ou RHEL est en cours d'exécution](#)
- [Vérifiez que l'CodeDeploy agent pour Ubuntu Server est en cours d'exécution](#)
- [Vérifiez que l'CodeDeploy agent pour Windows Server est en cours d'exécution](#)

Vérifiez que l' CodeDeploy agent pour Amazon Linux ou RHEL est en cours d'exécution

Pour vérifier si l' CodeDeploy agent est installé et en cours d'exécution, connectez-vous à l'instance et exécutez la commande suivante :

```
systemctl status codedeploy-agent
```

Si la commande renvoie une erreur, l' CodeDeploy agent n'est pas installé. Installez-le comme cela est décrit dans la rubrique [Installation de l' CodeDeploy agent pour Amazon Linux ou RHEL](#).

Si l' CodeDeploy agent est installé et en cours d'exécution, vous devriez voir un message tel que `The AWS CodeDeploy agent is running`.

Si vous voyez un message tel que `error: No AWS CodeDeploy agent running`, démarrez le service et exécutez les deux commandes suivantes, l'une après l'autre :

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Vérifiez que l' CodeDeploy agent pour Ubuntu Server est en cours d'exécution

Pour vérifier si l' CodeDeploy agent est installé et en cours d'exécution, connectez-vous à l'instance et exécutez la commande suivante :

```
systemctl status codedeploy-agent
```

Si la commande renvoie une erreur, l' CodeDeploy agent n'est pas installé. Installez-le comme cela est décrit dans la rubrique [Installation de l' CodeDeploy agent pour Ubuntu Server](#).

Si l' CodeDeploy agent est installé et en cours d'exécution, vous devriez voir un message tel que `The AWS CodeDeploy agent is running`.

Si vous voyez un message tel que `error: No AWS CodeDeploy agent running`, démarrez le service et exécutez les deux commandes suivantes, l'une après l'autre :

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Vérifiez que l' CodeDeploy agent pour Windows Server est en cours d'exécution

Pour vérifier si l' CodeDeploy agent est installé et en cours d'exécution, connectez-vous à l'instance et exécutez la commande suivante :

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Vous devez voir des résultats similaires à ce qui suit :

Status	Name	DisplayName
-----	----	-----
Running	codedeployagent	CodeDeploy Host Agent Service

Si la commande renvoie une erreur, l' CodeDeploy agent n'est pas installé. Installez-le comme cela est décrit dans la rubrique [Installation de l' CodeDeploy agent pour Windows Server](#).

Si Status montre quelque chose d'autre que Running, démarrez le service avec la commande suivante :

```
powershell.exe -Command Start-Service -Name codedeployagent
```

Vous pouvez redémarrer le service avec la commande suivante :

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Vous pouvez arrêter le service avec la commande suivante :

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Déterminer la version de l' CodeDeploy agent

Vous pouvez déterminer la version de l' CodeDeploy agent exécutée sur votre instance de deux manières.

Tout d'abord, à partir de la version 1.0.1.854 de l'agent CodeDeploy, vous pouvez afficher le numéro de version dans un `.version` fichier de l'instance. Le tableau suivant montre l'emplacement et l'exemple de chaîne de version pour chacun des systèmes d'exploitation pris en charge.

Système d'exploitation	Emplacement du fichier	Exemple de chaîne agent_version
Amazon Linux et Red Hat Enterprise Linux (RHEL)	<code>/opt/codedeploy-agent/.version</code>	<code>OFFICIAL_1.0.1.854_rpm</code>
Ubuntu Server	<code>/opt/codedeploy-agent/.version</code>	<code>OFFICIAL_1.0.1.854_deb</code>
Windows Server	<code>C:\ProgramData\Amazon\CodeDeploy\.version</code>	<code>OFFICIAL_1.0.1.854_msi</code>

Ensuite, vous pouvez exécuter une commande sur une instance afin de déterminer la version de l'agent CodeDeploy.

Rubriques

- [Déterminez la version sur Amazon Linux ou RHEL](#)
- [Déterminer la version sur le serveur Ubuntu](#)
- [Déterminer la version sur Windows Server](#)

Déterminez la version sur Amazon Linux ou RHEL

Connectez-vous à l'instance et exécutez la commande suivante :

```
sudo yum info codedeploy-agent
```

Déterminer la version sur le serveur Ubuntu

Connectez-vous à l'instance et exécutez la commande suivante :

```
sudo dpkg -s codedeploy-agent
```


Déterminer la version sur Windows Server

Connectez-vous à l'instance et exécutez la commande suivante :

```
sc qdescription codedeployagent
```

Installation de l' CodeDeploy agent

Pour être utilisé CodeDeploy sur des instances EC2 ou des serveurs locaux, l' CodeDeploy agent doit d'abord être installé. Nous vous recommandons d'installer et de mettre à jour l' CodeDeploy agent avec AWS Systems Manager. Pour plus d'informations sur Systems Manager, consultez [What is AWS Systems Manager](#). Vous pouvez configurer l'installation et les mises à jour planifiées de l' CodeDeploy agent avec Systems Manager dans la console lorsque vous créez vos groupes de déploiement.

Vous pouvez également installer l' CodeDeploy agent directement depuis un compartiment S3 à l'aide de la ligne de commande.

Pour connaître la version recommandée à installer, voir [Historique des versions de l' CodeDeploy agent](#).

Rubriques

- [Installez l' CodeDeploy agent à l'aide de AWS Systems Manager](#)
- [Installation de l' CodeDeploy agent à l'aide de la ligne de commande](#)

Installez l' CodeDeploy agent à l'aide de AWS Systems Manager

Vous pouvez utiliser le AWS Management Console ou AWS CLI pour installer l' CodeDeploy agent sur vos instances Amazon EC2 ou sur site en utilisant. AWS Systems Manager Vous pouvez choisir d'installer une version spécifique ou de toujours installer la version la plus récente de l'agent. Pour plus d'informations AWS Systems Manager, voir [Qu'est-ce que AWS Systems Manager](#).

L'utilisation AWS Systems Manager est la méthode recommandée pour installer et mettre à jour l' CodeDeploy agent. Vous pouvez également installer l' CodeDeploy agent à partir d'un compartiment Amazon S3. Pour de plus amples informations sur l'utilisation d'un lien de téléchargement Amazon S3, consultez [Installation de l' CodeDeploy agent à l'aide de la ligne de commande](#).

Rubriques

- [Conditions préalables](#)

- [Installation de l' CodeDeploy agent](#)

Conditions préalables

Suivez les étapes décrites [Commencer avec CodeDeploy](#) pour configurer les autorisations IAM et le AWS CLI.

Si vous installez l' CodeDeploy agent sur un serveur local avec Systems Manager, vous devez enregistrer votre serveur local auprès d'Amazon EC2 Systems Manager. Pour plus d'informations, consultez la section [Configuration de Systems Manager dans les environnements hybrides](#) dans le Guide de AWS Systems Manager l'utilisateur.

Installation de l' CodeDeploy agent

Avant de pouvoir utiliser Systems Manager pour installer l' CodeDeploy agent, vous devez vous assurer que l'instance est correctement configurée pour Systems Manager.

Installation ou mise à jour de l'agent SSM

Sur une instance Amazon EC2, l' CodeDeploy agent doit exécuter la version 2.3.274.0 ou ultérieure. Avant d'installer l' CodeDeploy agent, mettez à jour ou installez l'agent SSM sur l'instance si ce n'est déjà fait.

L'agent SSM est préinstallé sur certains Amazon AMIs EC2 fournis par. AWS Pour plus d'informations, consultez [Amazon Machine Images \(AMIs\) avec agent SSM préinstallé](#).

Note

Assurez-vous que le système d'exploitation de l'instance est également pris en charge par l' CodeDeploy agent. Pour de plus amples informations, veuillez consulter [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#).

Pour plus d'informations sur l'installation ou la mise à jour de l'agent SSM sur une instance exécutant Linux, consultez la section [Installation et configuration de l'agent SSM sur les instances Linux](#) dans le Guide de l'AWS Systems Manager utilisateur.

Pour plus d'informations sur l'installation ou la mise à jour de l'agent SSM sur une instance exécutant Windows Server, consultez la section [Installation et configuration de l'agent SSM sur les instances Windows](#) dans le Guide de l'AWS Systems Manager utilisateur.

(Facultatif) Vérification des prérequis de Systems Manager

Avant d'utiliser la commande Run Command de Systems Manager pour installer l' CodeDeploy agent, vérifiez que vos instances répondent aux exigences minimales de Systems Manager. Pour plus d'informations, consultez [Configuration d' AWS Systems Manager](#) dans le Guide de l'utilisateur AWS Systems Manager .

Installation de l' CodeDeploy agent

Avec SSM, vous pouvez l'installer CodeDeploy une seule fois ou établir un calendrier pour installer les nouvelles versions.

Pour installer l' CodeDeploy agent, choisissez le AWSCodeDeployAgent package en suivant les étapes décrites dans [Installer ou mettre à jour les packages avec le AWS Systems Manager distributeur](#).

Installation de l' CodeDeploy agent à l'aide de la ligne de commande

Note

Nous vous recommandons d'installer l' CodeDeploy agent AWS Systems Manager pour pouvoir configurer les mises à jour planifiées de l'agent. Pour de plus amples informations, veuillez consulter [Installez l' CodeDeploy agent à l'aide de AWS Systems Manager](#).

Utilisez les rubriques suivantes pour installer et exécuter l' CodeDeploy agent à l'aide de la ligne de commande.

Rubriques

- [Installation de l' CodeDeploy agent pour Amazon Linux ou RHEL](#)
- [Installation de l' CodeDeploy agent pour Ubuntu Server](#)
- [Installation de l' CodeDeploy agent pour Windows Server](#)

Installation de l' CodeDeploy agent pour Amazon Linux ou RHEL

Connectez-vous à l'instance et exécutez les commandes suivantes, l'une après l'autre. L'exécution de la commande en `sudo yum update` est considérée comme une bonne pratique lors de l'installation de packages, mais vous pouvez l'ignorer si vous ne souhaitez pas mettre à jour tous vos packages.

```
sudo yum update
```

```
sudo yum install ruby
```

```
sudo yum install wget
```

(Facultatif) Pour nettoyer l'AMI de toutes les informations de mise en cache de l'agent précédentes, exécutez le script suivant :

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Accédez à votre répertoire personnel :

```
cd /home/ec2-user
```

Note

Dans la commande précédente, `/home/ec2-user` représente le nom d'utilisateur par défaut pour une instance Amazon Linux ou RHEL Amazon EC2. Si votre instance a été créée à l'aide d'une image AMI personnalisée, le propriétaire de l'image AMI peut avoir spécifié un nom d'utilisateur par défaut différent.

Téléchargez le programme d'installation de l' CodeDeploy agent :

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

bucket-name est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région, ainsi *region-identifier* que l'identifiant de votre région.

Par exemple :

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
```

Pour obtenir la liste des noms de compartiments et des identificateurs de région, consultez [Noms des compartiments du kit de ressources par région](#).

Définissez les autorisations d'exécution sur le `install` fichier :

```
chmod +x ./install
```

Pour installer la dernière version de l' CodeDeploy agent :

- ```
sudo ./install auto
```

Pour installer une version spécifique de l' CodeDeploy agent :

- Répertoriez les versions disponibles dans votre région :

```
aws s3 ls s3://aws-codedeploy-region-identifiant/releases/ --region region-identifiant | grep '\.rpm$'
```

- Installez l'une des versions suivantes :

```
sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
```

#### Note

AWS prend en charge la dernière version mineure de l' CodeDeploy agent. Actuellement, la dernière version mineure est 1.7.x.

Pour vérifier que le service est en cours d'exécution, exécutez la commande suivante :

```
systemctl status codedeploy-agent
```

Si l' CodeDeploy agent est installé et en cours d'exécution, vous devriez voir un message tel que `The AWS CodeDeploy agent is running`.

Si vous voyez un message tel que `error: No AWS CodeDeploy agent running`, démarrez le service et exécutez les deux commandes suivantes, l'une après l'autre :

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

## Installation de l' CodeDeploy agent pour Ubuntu Server

### Note

Nous vous recommandons d'installer l' CodeDeploy agent AWS Systems Manager pour pouvoir configurer les mises à jour planifiées de l'agent. Pour de plus amples informations, veuillez consulter [Installez l' CodeDeploy agent à l'aide de AWS Systems Manager](#).

Pour installer l' CodeDeploy agent sur le serveur Ubuntu

1. Connectez-vous à l'instance.
2. Entrez les commandes suivantes, l'une après l'autre :

```
sudo apt update
```

```
sudo apt install ruby-full
```

```
sudo apt install wget
```

3. Entrez la commande suivante :

```
cd /home/ubuntu
```

*/home/ubuntu* représente le nom d'utilisateur par défaut pour une instance de serveur Ubuntu. Si votre instance a été créée à l'aide d'une image AMI personnalisée, le propriétaire de l'image AMI peut avoir spécifié un nom d'utilisateur par défaut différent.

4. Entrez la commande suivante :

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

*bucket-name* est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région, ainsi *region-identifier* que l'identifiant de votre région.

Par exemple :

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/
install
```

Pour obtenir la liste des noms de compartiments et des identificateurs de région, consultez [Noms des compartiments du kit de ressources par région](#).

5. Entrez la commande suivante :

```
chmod +x ./install
```

6. Effectuez l'une des actions suivantes :

- Pour installer la dernière version de l' CodeDeploy agent sur n'importe quelle version prise en charge d'Ubuntu Server à l'exception de la version 20.04 :

```
sudo ./install auto
```

- Pour installer la dernière version de l' CodeDeploy agent sur Ubuntu Server 20.04 :

 Note

L'écriture de la sortie dans un fichier journal temporaire est une solution de contournement qui doit être utilisée lorsque nous corrigeons un bogue connu avec le `install` script sur Ubuntu Server 20.04.

```
sudo ./install auto > /tmp/logfile
```

- Pour installer une version spécifique de l' CodeDeploy agent sur n'importe quelle version prise en charge d'Ubuntu Server à l'exception de la version 20.04 :
  - Répertoriez les versions disponibles dans votre région :

```
aws s3 ls s3://aws-codedeploy-region-identifiant/releases/ --region region-identifiant | grep '\.deb$'
```

- Installez l'une des versions suivantes :

```
sudo ./install auto -v releases/codedeploy-agent-###.deb
```

 Note

AWS prend en charge la dernière version mineure de l' CodeDeploy agent. Actuellement, la dernière version mineure est 1.7.x.

- Pour installer une version spécifique de l' CodeDeploy agent sur Ubuntu Server 20.04 :
  - Répertoriez les versions disponibles dans votre région :

```
aws s3 ls s3://aws-codedeploy-region-identifiant/releases/ --region region-identifiant | grep '\.deb$'
```

- Installez l'une des versions suivantes :

```
sudo ./install auto -v releases/codedeploy-agent-###.deb > /tmp/logfile
```

 Note

L'écriture de la sortie dans un fichier journal temporaire est une solution de contournement qui doit être utilisée lorsque nous corrigeons un bogue connu avec le `install` script sur Ubuntu Server 20.04.

 Note

AWS prend en charge la dernière version mineure de l' CodeDeploy agent. Actuellement, la dernière version mineure est 1.7.x.



## Pour vérifier que le service fonctionne

1. Entrez la commande suivante :

```
systemctl status codedeploy-agent
```

Si l' AWS CodeDeploy agent est installé et en cours d'exécution, vous devriez voir un message tel que `The AWS CodeDeploy agent is running.`

2. Si vous voyez un message tel que `error: No AWS CodeDeploy agent running`, démarrez le service et exécutez les deux commandes suivantes, l'une après l'autre :

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

## Installation de l' AWS CodeDeploy agent pour Windows Server

Sur les instances Windows Server, vous pouvez utiliser l'une des méthodes suivantes pour télécharger et installer l' AWS CodeDeploy agent :

- Utilisation AWS Systems Manager (recommandée)
- Exécutez une série de PowerShell commandes Windows.
- Choisissez un lien de téléchargement direct.
- Exécutez une commande de copie Amazon S3.

### Note

Le dossier dans lequel l' AWS CodeDeploy agent est installé est `C:\Program Data\Amazon\CodeDeploy`. Assurez-vous qu'il n'y a pas de jonctions de répertoires ou de liens symboliques sur ce chemin.

## Rubriques

- [Utiliser Systems Manager](#)
- [Utiliser Windows PowerShell](#)

- [Utilisation d'un lien direct](#)
- [Utiliser une commande de copie Amazon S3](#)

## Utiliser Systems Manager

Suivez les instructions [Installez l' CodeDeploy agent à l'aide de AWS Systems Manager](#) pour installer l' CodeDeployagent.

## Utiliser Windows PowerShell

Connectez-vous à l'instance et exécutez les commandes suivantes sous Windows PowerShell :

1. Exigez que tous les scripts et les fichiers de configuration téléchargés depuis Internet soient signés par un éditeur de confiance. Si vous êtes invité à modifier la stratégie d'exécution, tapez « Y ».

```
Set-ExecutionPolicy RemoteSigned
```

2. Chargez le AWS Tools for Windows PowerShell.

```
Import-Module AWSPowerShell
```

3. Créez un répertoire dans lequel le fichier d'installation de l' CodeDeploy agent est téléchargé.

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

4. Configurez AWS les informations d'identification à l'aide Set-AWSCredential des Initialize-AWSDefaultConfiguration commandes et. Pour plus d'informations, consultez la section [Utilisation des AWS informations d'identification](#) dans les AWS outils du Guide de PowerShell l'utilisateur.
5. Téléchargez le fichier d'installation de l' CodeDeploy agent.

### Note

AWS prend en charge la dernière version mineure de l' CodeDeploy agent. Actuellement, la dernière version mineure est 1.7.x.

Pour installer la dernière version de l' CodeDeployagent :

- ```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Pour installer une version spécifique de l' CodeDeploy agent :

- ```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
```

*bucket-name* est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio), remplacez *bucket-name* par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

6. Exécutez le fichier d'installation de l' CodeDeploy agent.

```
c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
```

Pour vérifier que le service est en cours d'exécution, exécutez la commande suivante :

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Si l' CodeDeploy agent vient d'être installé et n'a pas été démarré, après avoir exécuté la Get-Service commande, sous État, vous devriez voir **Start...** :

| Status   | Name            | DisplayName                   |
|----------|-----------------|-------------------------------|
| -----    | ----            | -----                         |
| Start... | codedeployagent | CodeDeploy Host Agent Service |

Si l' CodeDeploy agent est déjà en cours d'exécution, après avoir exécuté la Get-Service commande, sous État, vous devriez voir **Running** :

| Status  | Name            | DisplayName                   |
|---------|-----------------|-------------------------------|
| -----   | ----            | -----                         |
| Running | codedeployagent | CodeDeploy Host Agent Service |

## Utilisation d'un lien direct

Si les paramètres de sécurité du navigateur de l'instance Windows Server fournissent les autorisations (par exemple, pour `https://s3.*.amazonaws.com`), vous pouvez utiliser un lien direct vers votre région pour télécharger l' CodeDeploy agent, puis exécuter le programme d'installation manuellement.

Le lien est le suivant :

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... où se *region* trouve la AWS région dans laquelle vous déployez votre application.

Par exemple :

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latest/codedeploy-agent.msi
```

### Important

Procurez-vous le .msi fichier auprès de la même région que votre CodeDeploy demande. Le choix d'une autre région peut entraîner inconsistent region des défaillances dans le codedeploy-agent-log fichier lorsque vous l'.msi exécutez.

## Utiliser une commande de copie Amazon S3

S'il AWS CLI est installé sur l'instance, vous pouvez utiliser la commande Amazon S3 [cp](#) pour télécharger l' CodeDeploy agent, puis exécuter le programme d'installation manuellement. Pour plus d'informations, voir [Installer AWS Command Line Interface le sous Microsoft Windows](#).

La commande Amazon S3 est la suivante :

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
--region region
```

... où se *region* trouve la AWS région dans laquelle vous déployez votre application.

Par exemple :

```
aws s3 cp s3://aws-codedeploy-af-south-1/latest/codedeploy-agent.msi codedeploy-agent.msi --region af-south-1
```

## Mettre à jour l' CodeDeploy agent

Vous pouvez configurer des mises à jour automatiques et planifiées de l' CodeDeploy agent sur tous les systèmes d'exploitation pris en charge à l'aide de AWS Systems Manager. Vous pouvez également forcer les mises à jour sur tous les systèmes d'exploitation pris en charge en exécutant une commande sur une instance.

### Rubriques

- [Mettre à jour l' CodeDeploy agent sur Amazon Linux ou RHEL](#)
- [Mettre à jour l' CodeDeploy agent sur le serveur Ubuntu](#)
- [Mettre à jour l' CodeDeploy agent sur Windows Server](#)

## Mettre à jour l' CodeDeploy agent sur Amazon Linux ou RHEL

Pour configurer les mises à jour automatiques et planifiées de l' CodeDeploy agent à l'aide de AWS Systems Manager, suivez les étapes [décrites dans Installer l' CodeDeploy agent avec AWS Systems Manager](#).

Si vous souhaitez forcer la mise à jour de l' CodeDeploy agent, connectez-vous à l'instance et exécutez la commande suivante :

```
sudo /opt/codedeploy-agent/bin/install auto
```

## Mettre à jour l' CodeDeploy agent sur le serveur Ubuntu

Pour configurer les mises à jour automatiques et planifiées de l' CodeDeploy agent à l'aide de AWS Systems Manager, suivez les étapes [décrites dans Installer l' CodeDeploy agent avec AWS Systems Manager](#).

Si vous souhaitez forcer la mise à jour de l' CodeDeploy agent, connectez-vous à l'instance et exécutez la commande suivante :

```
sudo /opt/codedeploy-agent/bin/install auto
```

## Mettre à jour l' CodeDeploy agent sur Windows Server

Vous pouvez activer les mises à jour automatiques de l' CodeDeploy agent avec AWS Systems Manager. Avec Systems Manager, vous pouvez configurer un calendrier de mise à jour pour vos instances Amazon EC2 ou sur site en créant une association avec Systems Manager State Manager. Vous pouvez également mettre à jour l' CodeDeploy agent manuellement en désinstallant la version actuelle et en installant une version plus récente.

### Rubriques

- [Configurez la mise à jour automatique des CodeDeploy agents avec AWS Systems Manager](#)
- [Mettre à jour l' CodeDeploy agent manuellement](#)
- [\(Obsolète\) Mettez à jour l' CodeDeploy agent avec le Windows Server Updater](#)

Configurez la mise à jour automatique des CodeDeploy agents avec AWS Systems Manager

Pour configurer Systems Manager et activer les mises à jour automatiques de l' CodeDeploy agent, suivez les instructions de la section [Installer l' CodeDeploy agent à l'aide](#) de AWS Systems Manager.

### Mettre à jour l' CodeDeploy agent manuellement

Pour mettre à jour l' CodeDeploy agent manuellement, vous pouvez installer la dernière version à partir de la CLI ou à l'aide de Systems Manager. Suivez les instructions de [Installer l'agent CodeDeploy](#). Il est recommandé de désinstaller les anciennes versions de l' CodeDeploy agent en suivant les instructions de la section [Désinstaller l' CodeDeployagent](#).

(Obsolète) Mettez à jour l' CodeDeploy agent avec le Windows Server Updater

#### Note

Le programme de mise à jour de l' CodeDeploy agent pour Windows Server est obsolète et ne sera mis à jour vers aucune version postérieure à la version 1.0.1.1597.

Pour activer les mises à jour automatiques de l' CodeDeploy agent, installez le programme de mise à jour de l' CodeDeploy agent pour Windows Server sur des instances nouvelles ou existantes. Le programme de mise à jour vérifie régulièrement l'existence de nouvelles versions. Lorsqu'une nouvelle version est détectée, le programme de mise à jour désinstalle la version actuelle de l'agent, si elle est installée, avant d'installer la version la plus récente.

Si un déploiement est déjà en cours lorsque le programme de mise à jour détecte une nouvelle version, le déploiement se poursuit jusqu'à la fin. Si un déploiement tente de démarrer pendant le processus de mise à jour, le déploiement échoue.

Si vous souhaitez forcer la mise à jour de l' CodeDeploy agent, suivez les instructions figurant dans [Installation de l' CodeDeploy agent pour Windows Server](#).

Sur les instances Windows Server, vous pouvez télécharger et installer le programme de mise à jour de l' CodeDeploy agent en exécutant des PowerShell commandes Windows, en utilisant un lien de téléchargement direct ou en exécutant une commande de copie Amazon S3.

## Rubriques

- [Utiliser Windows PowerShell](#)
- [Utilisation d'un lien direct](#)
- [Utiliser une commande de copie Amazon S3](#)

## Utiliser Windows PowerShell

Connectez-vous à l'instance et exécutez les commandes suivantes sous Windows PowerShell, une par une :

```
Set-ExecutionPolicy RemoteSigned
```

Si vous êtes invité à modifier la politique d'exécution, choisissez de faire en sorte que Windows PowerShell exige que tous les scripts et fichiers de configuration téléchargés depuis Internet soient signés par un éditeur approuvé.

```
Import-Module AWSPowerShell
```

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/codedeploy-agent-updater.msi -File c:\temp\codedeploy-agent-updater.msi
```

```
c:\temp\codedeploy-agent-updater.msi /quiet /l c:\temp\host-agent-updater-log.txt
```

```
powershell.exe -Command Get-Service -Name codedeployagent
```

**bucket-name** est le nom du compartiment Amazon S3 qui contient les fichiers du kit de CodeDeploy ressources pour votre région. Par exemple, pour la région USA Est (Ohio), remplacez **bucket-name** par `aws-codedeploy-us-east-2`. Pour obtenir la liste des noms de compartiments, consultez [Noms des compartiments du kit de ressources par région](#).

Si vous devez résoudre une erreur du processus de mise à jour, tapez la commande suivante pour ouvrir le fichier journal du programme de mise à jour de l' CodeDeploy agent :

```
notepad C:\ProgramData\Amazon\CodeDeployUpdater\log\codedeploy-agent.updater.log
```

### Utilisation d'un lien direct

Si les paramètres de sécurité du navigateur de l'instance Windows Server fournissent les autorisations requises (par exemple, pour `http://s3.*.amazonaws.com` ), vous pouvez utiliser un lien direct pour télécharger le programme de mise à jour de l' CodeDeploy agent.

Le lien est le suivant :

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent-updater.msi
```

... où se **region** trouve la AWS région dans laquelle vous mettez à jour votre application.

Par exemple :

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latest/codedeploy-agent-updater.msi
```

### Utiliser une commande de copie Amazon S3

S'il AWS CLI est installé sur l'instance, vous pouvez utiliser la commande Amazon S3 [cp](#) pour télécharger le programme de mise à jour de l' CodeDeploy agent, puis exécuter le programme d'installation manuellement. Pour plus d'informations, voir [Installer AWS Command Line Interface le sous Microsoft Windows](#).

La commande Amazon S3 est la suivante :

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent-updater.msi codedeploy-agent-updater.msi --region region
```



... où se *region* trouve la AWS région dans laquelle vous mettez à jour votre application.

Par exemple :

```
aws s3 cp s3://aws-codedeploy-af-south-1/latest/codedeploy-agent-updater.msi
codedeploy-agent-updater.msi --region af-south-1
```

## Désinstallez l' CodeDeploy agent

Vous pouvez supprimer l' CodeDeploy agent des instances lorsqu'il n'est plus nécessaire ou lorsque vous souhaitez effectuer une nouvelle installation.

### Désinstallez l' CodeDeploy agent d'Amazon Linux ou de RHEL

Pour désinstaller l' CodeDeploy agent, connectez-vous à l'instance et exécutez la commande suivante :

```
sudo yum erase codedeploy-agent
```

### Désinstallez l' CodeDeploy agent d'Ubuntu Server

Pour désinstaller l' CodeDeploy agent, connectez-vous à l'instance et exécutez la commande suivante :

```
sudo dpkg --purge codedeploy-agent
```

### Désinstallez l' CodeDeploy agent de Windows Server

Pour désinstaller l' CodeDeploy agent, connectez-vous à l'instance et exécutez les trois commandes suivantes, une par une :

```
wmic
```

```
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
```

```
exit
```

Vous pouvez également vous connecter à l'instance et, dans le Panneau de configuration, ouvrir Programmes et fonctionnalités, choisir CodeDeploy Host Agent, puis choisir Désinstaller.

## Envoyer les journaux des CodeDeploy agents à CloudWatch

Vous pouvez envoyer les données métriques et de journal de l' CodeDeploy agent à CloudWatch l'aide de l' [CloudWatch agent unifié](#) ou, plus simplement, de l' CloudWatch agent.

Suivez les instructions ci-dessous pour installer l' CloudWatch agent et le configurer pour une utilisation avec CodeDeploy des agents.

### Conditions préalables

Avant de commencer, effectuez les tâches suivantes :

- Installez l' CodeDeploy agent et assurez-vous qu'il est en cours d'exécution. Pour plus d'informations, consultez [Installation de l' CodeDeploy agent](#) et [Vérifiez que l' CodeDeploy agent est en cours d'exécution](#).
- Installez l' CloudWatch agent. Pour plus d'informations, consultez la section [Installation de l' CloudWatch agent](#).
- Ajoutez les autorisations suivantes au profil d'instance CodeDeploy IAM :
  - CloudWatchLogsFullAccess
  - CloudWatchAgentServerPolicy

Pour plus d'informations sur le profil d' CodeDeploy instance, consultez [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#) la section de [Commencer avec CodeDeploy](#).

### Configuration de l' CloudWatch agent pour collecter les CodeDeploy journaux

Vous pouvez configurer l' CloudWatch agent en utilisant un assistant ou en créant ou en modifiant manuellement un fichier de configuration.

Pour configurer l' CloudWatch agent à l'aide de l'assistant (Linux)

1. Exécutez l'assistant, comme décrit dans [Exécuter l'assistant de configuration de l' CloudWatch agent](#).
2. Dans l'assistant, lorsque vous y êtes invité, Do you want to monitor any log files? entrez **1**.

3. Spécifiez le fichier journal de l' CodeDeploy agent, comme suit :
  1. Pour Log file path saisir le chemin du fichier CodeDeploy journal, par exemple :**/var/log/aws/codedeploy-agent/codedeploy-agent.log**.
  2. Pour Log group name saisir un nom de groupe de journaux, par exemple :**codedeploy-agent-log**.
  3. Pour Log stream name saisir un nom de flux de journal, par exemple :**{instance\_id}-codedeploy-agent-log**.
4. Lorsque vous y êtes invité Do you want to specify any additional log files?, entrez **1**.
5. Spécifiez les journaux de déploiement de l' CodeDeploy agent, comme suit :
  1. Pour Log file path saisir le chemin du fichier journal de CodeDeploy déploiement, par exemple :**/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log**.
  2. Pour Log group name saisir un nom de groupe de journaux, par exemple :**codedeploy-agent-deployment-log**.
  3. Pour Log stream name saisir un nom de flux de journal, par exemple :**{instance\_id}-codedeploy-agent-deployment-log**.
6. Lorsque vous y êtes invité Do you want to specify any additional log files?, entrez **1**.
7. Spécifiez les journaux du programme de mise à jour de l' CodeDeploy agent, comme suit :
  1. Pour Log file path saisir le chemin du fichier journal du CodeDeploy programme de mise à jour, par exemple :**/tmp/codedeploy-agent.update.log**.
  2. Pour Log group name saisir un nom de groupe de journaux, par exemple :**codedeploy-agent-updater-log**.
  3. Pour Log stream name saisir un nom de flux de journal, par exemple :**{instance\_id}-codedeploy-agent-updater-log**.

Pour configurer l' CloudWatch agent à l'aide de l'assistant (Windows)

1. Exécutez l'assistant, comme décrit dans [Exécuter l'assistant de configuration de l' CloudWatch agent](#).

2. Dans l'assistant, lorsque vous y êtes invité, Do you want to monitor any customized log files? entrez **1**.
3. Spécifiez le fichier CodeDeploy journal, comme suit :
  1. Pour Log file path saisir le chemin ou le fichier journal de l' CodeDeploy agent, par exemple : **C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt**.
  2. Pour Log group name saisir un nom de groupe de journaux, par exemple : **codedeploy-agent-log**.
  3. Pour Log stream name saisir un nom de flux de journal, par exemple : **{instance\_id}-codedeploy-agent-log**.
4. Lorsque vous y êtes invité Do you want to specify any additional log files?, entrez **1**.
5. Spécifiez les journaux de déploiement de l' CodeDeploy agent, comme suit :
  1. Pour Log file path saisir le chemin du fichier journal de CodeDeploy déploiement, par exemple : **C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log**.
  2. Pour Log group name saisir un nom de groupe de journaux, par exemple : **codedeploy-agent-deployment-log**.
  3. Pour Log stream name saisir un nom de flux de journal, par exemple : **{instance\_id}-codedeploy-agent-deployment-log**.

Pour configurer l' CloudWatch agent en créant ou en modifiant manuellement un fichier de configuration (Linux)

1. Créez ou modifiez le fichier de configuration de l' CloudWatch agent comme décrit dans [Création ou modification manuelle du fichier de configuration de l' CloudWatch agent](#).
2. Assurez-vous que le fichier est appelé /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json et qu'il contient le code suivant :

```
...
"logs": {
 "logs_collected": {
 "files": {
 "collect_list": [
 {
```

```

 "file_path": "/var/log/aws/codedeploy-agent/codedeploy-agent.log",
 "log_group_name": "codedeploy-agent-log",
 "log_stream_name": "{instance_id}-agent-log"
 },
 {
 "file_path": "/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log",
 "log_group_name": "codedeploy-agent-deployment-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-deployment-log"
 },
 {
 "file_path": "/tmp/codedeploy-agent.update.log",
 "log_group_name": "codedeploy-agent-updater-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-updater-log"
 }
]
}
...

```

Pour configurer l' CloudWatch agent en créant ou en modifiant manuellement un fichier de configuration (Windows)

1. Créez ou modifiez le fichier de configuration de l' CloudWatch agent comme décrit dans [Création ou modification manuelle du fichier de configuration de l' CloudWatch agent](#).
2. Assurez-vous que le fichier est appelé C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json et qu'il contient le code suivant :

```

...
"logs": {
 "logs_collected": {
 "files": {
 "collect_list": [
 {
 "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\\log\\codedeploy-agent-log.txt",
 "log_group_name": "codedeploy-agent-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-log"
 }
]
 }
 }
}

```

```
 },
 {
 "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\\
\\deployment-logs\\codedeploy-agent-deployments.log",
 "log_group_name": "codedeploy-agent-deployment-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-
deployment-log"
 }
]
},
...
}
},
...
```

## Redémarrer l' CloudWatch agent

Après avoir apporté vos modifications, redémarrez l' CloudWatch agent comme décrit dans [Démarrer l' CloudWatch agent](#).

# Utilisation d'instances pour CodeDeploy

CodeDeploy prend en charge les déploiements sur des instances exécutant Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) et Windows Server.

Vous pouvez l'utiliser CodeDeploy pour déployer à la fois sur des instances Amazon EC2 et sur des instances sur site. Une instance sur site est un appareil physique autre qu'une instance Amazon EC2 capable d'exécuter l'agent et de se connecter à CodeDeploy des points de terminaison de service AWS public. Vous pouvez l'utiliser CodeDeploy pour déployer simultanément une application sur des instances Amazon EC2 dans le cloud et sur un ordinateur de bureau PCs ou sur des serveurs de votre propre centre de données.

## Comparaison des instances Amazon EC2 avec des instances sur site

Le tableau suivant compare les instances Amazon EC2 et les instances sur site :

| Sujet                                                                                                                                                                                      | Instances Amazon EC2 | Instances sur site |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|--------------------|
| Vous devez installer et exécuter une version de l'CodeDeploy agent compatible avec le système d'exploitation exécuté sur l'instance.                                                       | Oui                  | Oui                |
| Exige que l'instance soit en mesure de se connecter au service CodeDeploy.                                                                                                                 | Oui                  | Oui                |
| Nécessite qu'un profil d'instance IAM soit attaché à l'instance. Le profil d'instance IAM doit être autorisé à participer aux CodeDeploy déploiements. Pour plus d'informations, consultez | Oui                  | Non                |

| Sujet                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Instances Amazon EC2 | Instances sur site |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|--------------------|
| <a href="#">Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2.</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                      |                    |
| <p>Nécessite que vous effectuiez l'une des opérations suivantes afin d'authentifier et d'enregistrer les instances :</p> <ul style="list-style-type: none"><li>• Créez un rôle IAM qui peut être assumé par un utilisateur IAM sur chaque instance afin de récupérer les informations d'identification temporaires régulièrement actualisées générées via. AWS Security Token Service</li><li>• Créez un utilisateur IAM pour chaque instance et stockez les informations d'identification du compte de l'utilisateur IAM en texte brut sur l'instance.</li></ul> | Non                  | Oui                |
| <p>Vous devez enregistrer chaque instance auprès de celle-ci CodeDeploy avant de pouvoir la déployer.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Non                  | Oui                |
| <p>Vous devez étiqueter chaque instance avant de CodeDeploy pouvoir la déployer.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Oui                  | Oui                |



| Sujet                                                                                                                                                                                                    | Instances Amazon EC2 | Instances sur site |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|--------------------|
| Peut participer aux scénarios Amazon EC2 Auto Scaling et Elastic Load Balancing dans le cadre CodeDeploy de déploiements.                                                                                | Oui                  | Non                |
| Peut être déployé à partir de compartiments et de GitHub référentiels Amazon S3.                                                                                                                         | Oui                  | Oui                |
| Peut prendre en charge les déclencheurs qui invitent à envoyer des notifications par SMS ou par e-mail lorsque des événements spécifiés se produisent en rapport avec les déploiements ou les instances. | Oui                  | Oui                |
| Est soumise à facturation pour les déploiements associés.                                                                                                                                                | Non                  | Oui                |

## Tâches d'instance pour CodeDeploy

Pour lancer ou configurer des instances à utiliser dans des déploiements, choisissez parmi les instructions suivantes :

Je souhaite lancer une nouvelle instance Amazon EC2 Amazon Linux ou Windows Server.

Pour lancer l'instance Amazon EC2 avec le moins d'effort possible, consultez. [Création d'une instance Amazon EC2 pour CodeDeploy \(modèle\)CloudFormation](#)

Pour lancer l'instance Amazon EC2 principal ement par vous-même, consultez. [Créez une](#)

[instance Amazon EC2 pour CodeDeploy \(AWS CLI ou une console Amazon EC2\)](#)

Je souhaite lancer un nouveau serveur Ubuntu ou une nouvelle instance RHEL Amazon EC2.

Consultez [Créez une instance Amazon EC2 pour CodeDeploy \(AWS CLI ou une console Amazon EC2\)](#).

Je souhaite configurer une instance Amazon Linux, Windows Server, Ubuntu Server ou RHEL Amazon EC2.

Consultez [Configurer une instance Amazon EC2 pour qu'elle fonctionne avec CodeDeploy](#).

Je souhaite configurer une instance Windows Server, Ubuntu Server ou RHEL sur site (appareils physiques qui ne sont pas des instances Amazon EC2).

Consultez [Working with On-Premises Instances](#).

Je souhaite CodeDeploy fournir un parc d'instances de remplacement lors d'un blue/green déploiement.

Consultez [Travailler avec des déploiements dans CodeDeploy](#).

Pour préparer les instances Amazon EC2 dans les groupes Amazon EC2 Auto Scaling, vous devez suivre certaines étapes supplémentaires. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).

## Rubriques

- [Tagging Instances for Deployments](#)
- [Working with Amazon EC2 Instances](#)
- [Working with On-Premises Instances](#)
- [View Instance Details](#)
- [Instance Health](#)

# Marquage des instances pour les groupes de déploiement dans CodeDeploy

Pour faciliter la gestion de vos instances Amazon EC2 et de vos instances sur site, vous pouvez utiliser des balises pour attribuer vos propres métadonnées à chaque ressource. Les balises vous permettent de classer vos instances de différentes manières (par exemple, par objectif, par propriétaire ou par environnement). Cette approche est utile lorsque vous avez un grand nombre d'instances. Vous pouvez rapidement identifier une instance ou un groupe d'instances en fonction des balises que vous lui avez affectées. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Pour plus d'informations, consultez [Marquage de vos ressources Amazon EC2](#).

Pour spécifier les instances incluses dans un groupe de CodeDeploy déploiement, vous devez spécifier des balises dans un ou plusieurs groupes de balises. Les instances qui répondent à vos critères de balise sont celles sur lesquelles est installée la dernière révision d'application au moment de la création d'un déploiement sur ce groupe de déploiement.

## Note

Vous pouvez également inclure des groupes Amazon EC2 Auto Scaling dans les groupes de déploiement, mais ils sont identifiés par leur nom plutôt que par des balises appliquées aux instances. Pour plus d'informations, consultez [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).

Les critères des instances d'un groupe de déploiement peuvent être aussi simples qu'une seule balise dans un même groupe de balises. Ils peuvent également être plus complexes, avec trois groupes de balises au plus contenant 10 balises chacun.

Si vous utilisez un seul groupe de balises, toute instance identifiée par au moins une balise de ce groupe est incluse dans le groupe de déploiement. Si vous utilisez plusieurs groupes de balises, seules sont incluses les instances identifiées par au moins une balise dans chacun des groupes de balises.

Les exemples suivants illustrent la façon dont les balises et les groupes de balises peuvent être utilisés pour sélectionner des instances pour un groupe de déploiement.

## Rubriques

- [Exemple 1 : groupe de balises unique, étiquette unique](#)
- [Exemple 2 : groupe de balises unique, balises multiples](#)
- [Exemple 3 : plusieurs groupes de balises, balises uniques](#)
- [Exemple 4 : plusieurs groupes de balises, plusieurs balises](#)

## Exemple 1 : groupe de balises unique, étiquette unique

Vous pouvez spécifier une seule balise dans un seul groupe de balises :

Groupe de tags 1

| Clé | Valeur         |
|-----|----------------|
| Nom | AppVersion-ABC |

Chaque instance à laquelle est ajoutée la balise Name=AppVersion-ABC fait partie du groupe de déploiement, même si d'autres balises lui sont appliquées.

CodeDeploy vue de configuration de la console :

☒ Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.  
**One tag group:** Any instance identified by the tag group will be deployed to.  
**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

Tag group 1

|                                   |                                             |
|-----------------------------------|---------------------------------------------|
| Key - <i>optional</i>             | Value - <i>optional</i>                     |
| <input type="text" value="Name"/> | <input type="text" value="AppVersion-ABC"/> |
| <input type="text"/>              | <input type="text"/>                        |

Structure JSON :

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "key": "Name",
 "value": "AppVersion-ABC"
 }
]
]
}
```

```
{
 "Type": "KEY_AND_VALUE",
 "Key": "Name",
 "Value": "AppVersion-ABC"
}
],
},
```

## Exemple 2 : groupe de balises unique, balises multiples

Vous pouvez également spécifier plusieurs balises dans un seul groupe de balises :

Groupe de tags 1

| Clé    | Valeur |
|--------|--------|
| Région | Nord   |
| Région | Sud    |
| Région | Est    |

Une instance à laquelle est ajoutée l'une de ces trois balises fait partie du groupe de déploiement, même si d'autres balises lui sont appliquées. Par exemple, si vous disposez d'autres instances auxquelles est ajoutée la balise `Region=West`, elles ne sont pas incluses dans le groupe de déploiement.

CodeDeploy vue de configuration de la console :

☒ Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - *optional*Value - *optional*

|                                     |                                    |                                           |
|-------------------------------------|------------------------------------|-------------------------------------------|
| <input type="text" value="Region"/> | <input type="text" value="North"/> |                                           |
| <input type="text" value="Region"/> | <input type="text" value="South"/> | <input type="button" value="Remove tag"/> |
| <input type="text" value="Region"/> | <input type="text" value="East"/>  | <input type="button" value="Remove tag"/> |

Structure JSON :

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "South"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "East"
 }
]
]
},
```

## Exemple 3 : plusieurs groupes de balises, balises uniques

Vous pouvez également utiliser plusieurs ensembles de groupes de balises comprenant chacun une seule paire clé-valeur pour spécifier les critères des instances d'un groupe de déploiement. Lorsque vous utilisez plusieurs groupes de balises dans un groupe de déploiement, seules les instances identifiées par tous les groupes de balises sont incluses dans le groupe de déploiement.

### Groupe de tags 1

| Clé | Valeur         |
|-----|----------------|
| Nom | AppVersion-ABC |

### Groupe de tags 2

| Clé    | Valeur |
|--------|--------|
| Région | Nord   |

### Groupe de tags 3

| Clé  | Valeur    |
|------|-----------|
| Type | t2.medium |

Vous pouvez disposer de différents types d'instances dans plusieurs régions comportant la balise `Name=AppVersion-ABC`. Dans cet exemple, seules les instances comportant également les balises `Region=North` et `Type=t2.medium` font partie du groupe de déploiement.

CodeDeploy vue de configuration de la console :

☒ Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - *optional*Value - *optional*

## Tag group 2

Key - *optional*Value - *optional*

## Tag group 3

Key - *optional*Value - *optional*

## Structure JSON :

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Name",
 "Value": "AppVersion-ABC"
 }
]
],
}
```



```
[
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.medium"
 }
],
],
},
```

## Exemple 4 : plusieurs groupes de balises, plusieurs balises

Lorsque vous utilisez plusieurs groupes de balises avec plusieurs balises dans un ou plusieurs groupes, une instance doit correspondre à au moins l'une des balises de chacun des groupes.

### Groupe de tags 1

| Clé           | Valeur        |
|---------------|---------------|
| Environnement | Bêta          |
| Environnement | Intermédiaire |

### Groupe de tags 2

| Clé    | Valeur |
|--------|--------|
| Région | Nord   |
| Région | Sud    |
| Région | Est    |

## Groupe de tags 3

| Clé  | Valeur    |
|------|-----------|
| Type | t2.medium |
| Type | t2.large  |

Dans cet exemple, pour pouvoir être incluse dans le groupe de déploiement, une instance doit comporter les balises (1) `Environment=Beta` ou `Environment=Staging`, (2) `Region=North` ou `Region=South` ou `Region=East` et (3) `Type=t2.medium` ou `Type=t2.large`.

Ainsi, des instances comportant les groupes de balises suivants seraient incluses dans le groupe de déploiement :

- `Environment=Beta, Region=North, Type=t2.medium`
- `Environment=Staging, Region=East, Type=t2.large`
- `Environment=Staging, Region=South, Type=t2.large`

À l'inverse, des instances comportant les groupes de balises suivants ne seraient pas incluses dans le groupe de déploiement. Les valeurs de clé en surbrillance entraînent l'exclusion des instances :

- `Environment=Beta, Région=Ouest, Type=t2.medium`
- `Environment=Staging, Region=East, Type=t2.micro`
- `Environnement=Production, Region=South, Type=t2.large`

CodeDeploy vue de configuration de la console :

☒ Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - *optional*Value - *optional*

|                 |             |
|-----------------|-------------|
| Q Environment X | Q Staging X |
|-----------------|-------------|

|                 |          |            |
|-----------------|----------|------------|
| Q Environment X | Q Beta X | Remove tag |
|-----------------|----------|------------|

Add tag

## Tag group 2

Key - *optional*Value - *optional*

|            |           |
|------------|-----------|
| Q Region X | Q South X |
|------------|-----------|

|            |           |            |
|------------|-----------|------------|
| Q Region X | Q North X | Remove tag |
|------------|-----------|------------|

|            |          |            |
|------------|----------|------------|
| Q Region X | Q East X | Remove tag |
|------------|----------|------------|

Add tag↶ Remove tag group

## Tag group 3

Key - *optional*Value - *optional*

|          |               |
|----------|---------------|
| Q Type X | Q t2.medium X |
|----------|---------------|

|          |              |            |
|----------|--------------|------------|
| Q Type X | Q t2.large X | Remove tag |
|----------|--------------|------------|

Add tag↶ Remove tag group

## Structure JSON :

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Environment",
 "Value": "Beta"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Environment",
 "Value": "Staging"
 }
],
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "South"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "East"
 }
],
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.medium"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.large"
 }
]
]
}
```

```
],
],
},
```

## Utilisation d'instances Amazon EC2 pour CodeDeploy

Une instance Amazon EC2 est un environnement informatique virtuel que vous créez et configurez à l'aide d'Amazon Elastic Compute Cloud. Amazon EC2 fournit une capacité de calcul évolutive dans le AWS cloud. Vous pouvez utiliser Amazon EC2 pour lancer autant ou aussi peu de serveurs virtuels que nécessaire pour vos CodeDeploy déploiements.

Pour plus d'informations sur Amazon EC2, consultez le guide de [démarrage Amazon EC2](#).

Les instructions de cette section vous montrent comment créer et configurer des instances Amazon EC2 à utiliser dans vos CodeDeploy déploiements.

### Rubriques

- [Créez une instance Amazon EC2 pour CodeDeploy \(AWS CLI ou une console Amazon EC2\)](#)
- [Création d'une instance Amazon EC2 pour CodeDeploy \(modèle\)CloudFormation](#)
- [Configurer une instance Amazon EC2 pour qu'elle fonctionne avec CodeDeploy](#)

## Créez une instance Amazon EC2 pour CodeDeploy (AWS CLI ou une console Amazon EC2)

Ces instructions vous montrent comment lancer une nouvelle instance Amazon EC2 configurée pour être utilisée dans CodeDeploy des déploiements.

Vous pouvez utiliser notre CloudFormation modèle pour lancer une instance Amazon EC2 exécutant Amazon Linux ou Windows Server qui est déjà configurée pour être utilisée dans CodeDeploy des déploiements. Nous ne fournissons pas de CloudFormation modèle pour les instances Amazon EC2 exécutant Ubuntu Server ou Red Hat Enterprise Linux (RHEL). Pour découvrir des alternatives à l'utilisation du modèle, consultez la page [Utilisation d'instances pour CodeDeploy](#).

Vous pouvez utiliser la console Amazon EC2 ou les AWS CLI API Amazon EC2 pour lancer une instance Amazon EC2.

## Lancer une instance Amazon EC2 (console)

### Conditions préalables

Si ce n'est pas déjà fait, suivez les instructions fournies [Commencer avec CodeDeploy](#) pour configurer AWS CLI et créer un profil d'instance IAM.

### Lancement d'une instance Amazon EC2

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
2. Dans le volet de navigation, choisissez Instances, puis Lancer une instance.
3. Sur la page Étape 1 : Sélection d'une Amazon Machine Image (AMI), à partir de l'onglet Quick Start, recherchez le système d'exploitation et la version que vous souhaitez utiliser, puis choisissez Sélectionner. Vous devez choisir un système d'exploitation AMI Amazon EC2 pris en charge par. CodeDeploy Pour de plus amples informations, veuillez consulter [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#).
4. Sur la page Étape 2 : Choisissez un type d'instance, choisissez n'importe quel type d'instance Amazon EC2 disponible, puis choisissez Suivant : Configurer les détails de l'instance.
5. Sur la page Étape 3 : Configuration des détails de l'instance, dans la liste des rôles IAM, choisissez le rôle d'instance IAM que vous avez créé dans. [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#) Si vous avez utilisé le nom de rôle suggéré, choisissez **CodeDeployDemo-EC2-Instance-Profile**. Si vous avez créé votre propre nom de rôle, choisissez-le.

#### Note

Si un cloud privé virtuel (VPC) par défaut ne figure pas dans la liste des réseaux, vous devez choisir ou créer un VPC et un sous-réseau Amazon. Choisissez Create new VPC (Créer un VPC) ou Create new subnet (Créer un sous-réseau), ou les deux. Pour plus d'informations, consultez [Votre VPC et](#) vos sous-réseaux.

6. Choisissez Next: Add Storage (Suivant : Ajouter le stockage).
7. Laissez la page Step 4: Add Storage inchangée, puis choisissez Next: Add Tags.
8. Sur la page Step 5: Add Tags (Étape 5 : Ajouter des balises), choisissez Add Tag (Ajouter une balise).
9. Dans la zone Clé, tapez **Name**. Dans la zone Value, tapez **CodeDeployDemo**.

 Important

Le contenu des zones Clé et Valeur est sensible à la casse.

10. Choisissez Suivant : Configurer le groupe de sécurité.
11. Sur la page Étape 6 : Configurer le groupe de sécurité, laissez l'option Créez un nouveau groupe de sécurité sélectionnée.

Un rôle SSH par défaut est configuré pour les instances Amazon EC2 exécutant Amazon Linux, Ubuntu Server ou RHEL. Un rôle RDP par défaut est configuré pour les instances Amazon EC2 exécutant Windows Server.

12. Si vous souhaitez ouvrir le port HTTP, choisissez le bouton Ajouter une règle et, dans la liste déroulante Type, choisissez **HTTP**. Acceptez la valeur Source par défaut de Custom 0.0.0.0/0, puis choisissez Review and Launch.

 Note

Dans un environnement de production, nous recommandons de restreindre l'accès aux ports SSH, RDP et HTTP, au lieu de spécifier Anywhere 0.0.0.0/0. CodeDeploy ne nécessite pas d'accès illimité aux ports et ne nécessite pas d'accès HTTP. Pour plus d'informations, consultez la section [Conseils pour sécuriser votre instance Amazon EC2](#).

Si une boîte de dialogue Démarrage depuis General Purpose (SSD) apparaît, suivez les instructions, puis choisissez Suivant.

13. Laissez la page Étape 7 : Examiner le lancement de l'instance inchangée et choisissez Lancement.
14. Dans la boîte de dialogue Sélectionnez une paire de clés (key pair) existante ou créez une nouvelle paire de clés (key pair), choisissez Choisir une paire de clés (key pair) existante ou Créer une nouvelle paire de clés (key pair). Si vous avez déjà configuré une paire de clés d'instance Amazon EC2, vous pouvez la choisir ici.

Si vous ne disposez pas déjà d'une paire de clés d'instance Amazon EC2, choisissez Create a new key pair (Créer une nouvelle paire de clés) et attribuez-lui un nom facilement identifiable. Choisissez Download Key Pair pour télécharger la paire de clés d'instance Amazon EC2 sur votre ordinateur.

 Important

Vous devez disposer d'une paire de clés si vous souhaitez accéder à votre instance Amazon EC2 via SSH ou RDP.

15. Choisissez Launch Instances (Démarrer les instances).
16. Choisissez l'ID de votre instance Amazon EC2. Ne continuez pas tant que l'instance n'a pas été lancée et n'a pas réussi tous les contrôles.

## Installation de l' CodeDeploy agent

L' CodeDeploy agent doit être installé sur votre instance Amazon EC2 avant de l'utiliser dans CodeDeploy des déploiements. Pour de plus amples informations, veuillez consulter [Installation de l' CodeDeploy agent](#).

 Note

Vous pouvez configurer l'installation automatique et les mises à jour de l' CodeDeploy agent lorsque vous créez votre groupe de déploiement dans la console.

## Lancer une instance Amazon EC2 (CLI)

### Conditions préalables

Si ce n'est pas déjà fait, suivez les instructions fournies [Commencer avec CodeDeploy](#) pour configurer AWS CLI et créer un profil d'instance IAM.

### Lancement d'une instance Amazon EC2

1. Pour Windows Server uniquement Si vous créez une instance Amazon EC2 exécutant Windows Server, appelez les authorize-security-group-ingress commandes create-security-group and pour créer un groupe de sécurité qui autorise l'accès RDP (qui n'est pas autorisé par défaut) et, sinon, l'accès HTTP. Par exemple, pour créer un groupe de sécurité nommé CodeDeployDemo-Windows-Security-Group, exécutez les commandes suivantes, une par une :

```
aws ec2 create-security-group --group-name CodeDeployDemo-Windows-Security-Group --description "For launching Windows Server images for use with CodeDeploy"
```



```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-Security-Group --to-port 3389 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port 3389
```

```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-Security-Group --to-port 80 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port 80
```

### Note

A des fins de démonstration, ces commandes créent un groupe de sécurité qui autorise un accès illimité pour RDP via le port 3389 et pour HTTP, via le port 80. Comme bonne pratique, nous vous recommandons de limiter l'accès aux ports RDP et HTTP. CodeDeploy requiert pas un accès illimité aux ports et ne requiert pas l'accès HTTP. Pour plus d'informations, consultez la section [Conseils pour sécuriser votre instance Amazon EC2](#).

## 2. Appelez la `run-instances` commande pour créer et lancer l'instance Amazon EC2.

Avant d'appeler cette commande, vous devez collecter les éléments suivants :

- L'ID d'une Amazon Machine Image (AMI) (*ami-id*) que vous utilisez pour l'instance. Pour obtenir l'ID, consultez la section [Trouver une AMI appropriée](#).
- Le nom du type d'instance Amazon EC2 (*instance-type*) que vous créez, tel que `t1.micro`. Pour obtenir une liste, consultez la section Types d'[instances Amazon EC2](#).
- Le nom d'un profil d'instance IAM autorisé à accéder au compartiment Amazon S3 dans lequel sont stockés les fichiers d'installation de l' CodeDeploy agent pour votre région.

Pour plus d'informations sur la création d'un profil d'instance IAM, consultez [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).

- Le nom d'une paire de clés d'instance Amazon EC2 (*key-name*) permettant l'accès SSH à une instance Amazon EC2 exécutant Amazon Linux, Ubuntu Server, RHEL ou RDP à une instance Amazon EC2 exécutant Windows Server.

 Important

Tapez le nom de la paire de clés uniquement, pas l'extension de fichier de la paire de clés. Par exemple, my-keypair, et non my-keypair.pem.

Pour trouver le nom d'une paire de clés, ouvrez la console Amazon EC2 à <https://console.aws.amazon.com/ec2/>. Dans le volet de navigation, sous Réseau et sécurité, choisissez Paires de clés et notez le nom de la paire de clés dans la liste.

Pour générer une paire de clés, consultez [Création de votre paire de clés à l'aide d'Amazon EC2](#). Assurez-vous de créer la paire de clés dans l'une des régions répertoriées dans [Région et les points de terminaison](#) dans Références générales AWS. Dans le cas contraire, vous ne pourrez pas utiliser la paire de clés de l'instance Amazon EC2 avec CodeDeploy.

Pour Amazon Linux, RHEL et Ubuntu Server

Pour appeler la `run-instances` commande pour lancer une instance Amazon EC2 exécutant Amazon Linux, Ubuntu Server ou RHEL et attacher le profil d'instance IAM que vous avez créé dans [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#) Par exemple :

```
aws ec2 run-instances \
 --image-id ami-id \
 --key-name key-name \
 --count 1 \
 --instance-type instance-type \
 --iam-instance-profile Name=iam-instance-profile
```

 Note

Cette commande crée un groupe de sécurité par défaut pour l'instance Amazon EC2 qui permet d'accéder à plusieurs ports, y compris un accès illimité pour le SSH via le port 22 ou, sinon, pour le protocole HTTP via le port 80. En tant que bonne pratique, nous recommandons de restreindre l'accès aux ports SSH et HTTP uniquement. CodeDeploy ne nécessite pas d'accès illimité au port et ne nécessite pas d'accès au port HTTP. Pour

plus d'informations, consultez la section [Conseils pour sécuriser votre instance Amazon EC2](#).

## Pour Windows Server

Pour appeler la `run-instances` commande pour lancer une instance Amazon EC2 exécutant Windows Server, joindre le profil d'instance IAM que vous avez créé et spécifier le nom du groupe de sécurité que vous avez créé à l'étape 1. [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#) Par exemple :

```
aws ec2 run-instances --image-id ami-id --key-name key-name --count 1 --instance-type instance-type --iam-instance-profile Name=iam-instance-profile --security-groups CodeDeploy-Windows-Security-Group
```

Ces commandes lancent une instance Amazon EC2 unique avec l'AMI, la paire de clés et le type d'instance spécifiés, avec le profil d'instance IAM spécifié, et exécutent le script spécifié lors du lancement.

3. Notez la valeur de `InstanceId` dans la sortie. Si vous oubliez cette valeur, vous pouvez l'obtenir ultérieurement en appelant la `describe-instances` commande sur la paire de clés de l'instance Amazon EC2.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query "Reservations[*].Instances[*].[InstanceId]" --output text
```

Utilisez l'ID d'instance pour appeler la `create-tags` commande, qui balise l'instance Amazon EC2 afin de CodeDeploy pouvoir la retrouver ultérieurement lors d'un déploiement. Dans l'exemple suivant, la balise est nommée **CodeDeployDemo**, mais vous pouvez spécifier n'importe quelle balise d'instance Amazon EC2 de votre choix.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

Vous pouvez appliquer plusieurs balises simultanément à une instance. Par exemple :

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance Key=Region,Value=West Key=Environment,Value=Beta
```

Pour vérifier que l'instance Amazon EC2 a été lancée et a passé toutes les vérifications, utilisez l'ID de l'instance pour appeler la `describe-instance-status` commande.

```
aws ec2 describe-instance-status --instance-ids instance-id --query
"InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Si l'instance a été lancée et a réussi tous les contrôles, ok s'affiche dans la sortie :

### Installation de l' CodeDeploy agent

L' CodeDeploy agent doit être installé sur votre instance Amazon EC2 avant de l'utiliser dans CodeDeploy des déploiements. Pour de plus amples informations, veuillez consulter [Installation de l' CodeDeploy agent](#).

#### Note

Vous pouvez configurer l'installation automatique et les mises à jour de l' CodeDeploy agent lorsque vous créez votre groupe de déploiement dans la console.

## Création d'une instance Amazon EC2 pour CodeDeploy (modèle)CloudFormation

Vous pouvez utiliser notre CloudFormation modèle pour lancer rapidement une instance Amazon EC2 exécutant Amazon Linux ou Windows Server. Vous pouvez utiliser AWS CLI la CodeDeploy console ou le AWS APIs pour lancer l'instance avec le modèle. Outre le lancement de l'instance, le modèle effectue les opérations suivantes :

- Indique d' CloudFormation autoriser l'instance à participer aux CodeDeploy déploiements.
- Marque l'instance afin de CodeDeploy pouvoir la retrouver lors d'un déploiement.
- Installe et exécute l' CodeDeploy agent sur l'instance.

Vous n'êtes pas obligé d'utiliser notre CloudFormation pour configurer une instance Amazon EC2. Pour découvrir des alternatives, consultez la page [Utilisation d'instances pour CodeDeploy](#).

Nous ne fournissons pas de CloudFormation modèle pour les instances Amazon EC2 exécutant Ubuntu Server ou Red Hat Enterprise Linux (RHEL).

## Rubriques

- [Avant de commencer](#)
- [Lancer une instance Amazon EC2 avec le CloudFormation modèle \(console\)](#)
- [Lancez une instance Amazon EC2 avec le CloudFormation modèle \(\)AWS CLI](#)

## Avant de commencer

Avant de pouvoir utiliser le CloudFormation modèle pour lancer des instances Amazon EC2, assurez-vous de suivre les étapes suivantes.

1. Assurez-vous d'avoir créé un utilisateur administratif, comme décrit dans [Étape 1 : Configuration](#). Vérifiez que l'utilisateur dispose des autorisations minimales suivantes et ajoutez celles qui ne sont pas présentes :
  - formation des nuages : \*
  - codedeploy:\*
  - ec2:\*
  - iam : AddRoleToInstanceProfile
  - iam : CreateInstanceProfile
  - iam : CreateRole
  - iam : DeleteInstanceProfile
  - iam : DeleteRole
  - iam : DeleteRolePolicy
  - iam : GetRole
  - iam : DeleteRolePolicy
  - iam : PutRolePolicy
  - iam : RemoveRoleFromInstanceProfile
2. Assurez-vous de disposer d'une paire de clés d'instance pour activer l'accès SSH à l'instance Amazon EC2 exécutant Amazon Linux ou l'accès RDP à l'instance exécutant Windows Server.

[Pour trouver le nom d'une paire de clés, ouvrez la console Amazon EC2 à https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/). Dans le volet de navigation, sous Réseau et sécurité, choisissez Paires de clés et notez le nom de la paire de clés dans la liste.

Pour générer une nouvelle paire de clés, consultez [Création de votre paire de clés à l'aide d'Amazon EC2](#). Assurez-vous que la paire de clés est créée dans l'une des régions répertoriées dans [Région et que les points de terminaison](#) sont créés dans Références générales AWS. Sinon, vous ne pourrez pas utiliser la paire de clés de l'instance avec CodeDeploy.

## Lancer une instance Amazon EC2 avec le CloudFormation modèle (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.

### Important

Connectez-vous au AWS Management Console avec le même compte que celui que vous avez utilisé [Commencer avec CodeDeploy](#). Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et des points de terminaison](#) dans. Références générales AWS CodeDeploy ne prend en charge que ces régions.

2. Sélectionnez Créer une pile.
3. Dans Choisir un modèle, choisissez Spécifier l'URL d'un modèle Amazon S3. Dans le champ, tapez l'emplacement du CloudFormation modèle pour votre région, puis choisissez Next.

| Région                         | Emplacement du CloudFormation modèle                                                                                       |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Région US East (Ohio)          | <code>http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région US East (N. Virginia)   | <code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Région US West (N. California) | <code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-we</code>                                                        |

| Région                    | Emplacement du CloudFormation modèle                                                                                |
|---------------------------|---------------------------------------------------------------------------------------------------------------------|
|                           | st-1/templates/latest/CodeDeploy_SampleCF_Template.json                                                             |
| Région US West (Oregon)   | http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json       |
| Région Canada (Centre)    | http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json |
| Région Europe (Irlande)   | http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json       |
| Région Europe (Londres)   | http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json       |
| Région Europe (Paris)     | http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json       |
| Région Europe (Francfort) | http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json |

| Région                            | Emplacement du CloudFormation modèle                                                                                                 |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Région Israël (Tel Aviv)          | <code>http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Région Asie-Pacifique (Hong Kong) | <code>http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Asia Pacific (Tokyo) Region       | <code>http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Asia Pacific (Seoul) Region       | <code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Asie-Pacifique (Singapour) | <code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Asia Pacific (Sydney)      | <code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code> |



| Région                             | Emplacement du CloudFormation modèle                                                                                                  |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Région Asie-Pacifique (Melbourne)  | <code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Asie-Pacifique (Mumbai)     | <code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>          |
| Région Amérique du Sud (São Paulo) | <code>aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code>              |

4. Dans la zone Stack name (Nom de pile), saisissez un nom pour la pile (par exemple **CodeDeployDemoStack**).
5. Dans Paramètres, saisissez ce qui suit et choisissez Suivant.
  - Pour InstanceCount, saisissez le nombre d'instances que vous souhaitez lancer. (Nous vous recommandons de conserver la valeur par défaut 1.)
  - Pour InstanceType, tapez le type d'instance que vous souhaitez lancer (ou laissez la valeur par défaut t1.micro).
  - Pour KeyPairName, tapez le nom de la paire de clés de l'instance. Tapez le nom de la paire de clés uniquement, pas l'extension de fichier de la paire de clés.
  - Dans le OperatingSystem champ box, tapez **Windows** pour lancer des instances exécutant Windows Server (ou conservez la valeur par défaut de Linux).
  - Pour SSHLocation, tapez la plage d'adresses IP à utiliser pour la connexion à l'instance via SSH ou RDP (ou conservez la valeur par défaut de 0.0.0.0/0).

 Important

La valeur par défaut **0.0.0.0/0** est fournie à des fins de démonstration uniquement. CodeDeploy n'exige pas que les instances Amazon EC2 aient un accès illimité aux ports. Comme bonne pratique, nous vous recommandons de limiter l'accès aux ports SSH (et HTTP). Pour plus d'informations, consultez la section [Conseils pour sécuriser votre instance Amazon EC2](#).

- Pour TagKey, tapez la clé de balise d'instance CodeDeploy qui sera utilisée pour identifier les instances lors du déploiement (ou laissez la valeur par défaut Nom).
  - Pour TagValue, tapez la valeur de balise d'instance CodeDeploy qui sera utilisée pour identifier les instances lors du déploiement (ou conservez la valeur par défaut de CodeDeployDemo).
6. Dans la page Options, laissez les zones d'option vides et sélectionnez Suivant.

 Important

CloudFormation les tags sont différents des CodeDeploy tags. CloudFormation utilise des balises pour simplifier l'administration de votre infrastructure. CodeDeploy utilise des balises pour identifier les instances Amazon EC2. Vous avez spécifié des balises CodeDeploy sur la page Specify Parameters (Spécifier les paramètres).

7. Sur la page de révision, dans Fonctionnalités, cochez la case Je reconnais que cela CloudFormation pourrait créer des ressources IAM, puis choisissez Créer.

Après avoir CloudFormation créé la pile et lancé les instances Amazon EC2, CREATE\_COMPLETE sera affiché dans la colonne Status de la CloudFormation console. Ce processus peut prendre plusieurs minutes.

Pour vérifier que l' CodeDeploy agent s'exécute sur les instances Amazon EC2, consultez [Gestion des opérations des CodeDeploy agents](#), puis passez à. [Créez une application avec CodeDeploy](#)

## Lancez une instance Amazon EC2 avec le CloudFormation modèle ()AWS CLI

1. Utilisez notre CloudFormation modèle lors d'un appel à la create-stack commande. Cette pile lancera une nouvelle instance Amazon EC2 avec l' CodeDeploy agent installé.

Pour lancer une instance Amazon EC2 exécutant Amazon Linux :

```
aws cloudformation create-stack \
 --stack-name CodeDeployDemoStack \
 --template-url templateURL \
 --parameters ParameterKey=InstanceCount,ParameterValue=1
ParameterKey=InstanceType,ParameterValue=t1.micro \
ParameterKey=KeyPairName,ParameterValue=keyName \
ParameterKey=OperatingSystem,ParameterValue=Linux \
ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0 \
ParameterKey=TagKey,ParameterValue=Name \
ParameterKey=TagValue,ParameterValue=CodeDeployDemo \
 --capabilities CAPABILITY_IAM
```

Pour lancer une instance Amazon EC2 exécutant Windows Server :

```
aws cloudformation create-stack --stack-name CodeDeployDemoStack --template-
url template-url --parameters ParameterKey=InstanceCount,ParameterValue=1
ParameterKey=InstanceType,ParameterValue=t1.micro
ParameterKey=KeyPairName,ParameterValue=keyName
ParameterKey=OperatingSystem,ParameterValue=Windows
ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
ParameterKey=TagKey,ParameterValue=Name
ParameterKey=TagValue,ParameterValue=CodeDeployDemo --capabilities CAPABILITY_IAM
```

*keyName* est le nom de la paire de clés de l'instance. Tapez le nom de la paire de clés uniquement, pas l'extension de fichier de la paire de clés.

*template-url* est l'emplacement du CloudFormation modèle pour votre région :

| Région                | Emplacement du CloudFormation modèle                                                                                                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Région US East (Ohio) | <a href="http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json">http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json</a> |

| Région                         | Emplacement du CloudFormation modèle                                                                                             |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Région US East (N. Virginia)   | <code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>                 |
| Région US West (N. California) | <code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Région US West (Oregon)        | <code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Région Canada (Centre)         | <code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Europe (Irlande)        | <code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Région Europe (Londres)        | <code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Région Europe (Paris)          | <code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |

| Région                            | Emplacement du CloudFormation modèle                                                                                                 |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Région Europe (Francfort)         | <code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Région Israël (Tel Aviv)          | <code>http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Région Asie-Pacifique (Hong Kong) | <code>http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Asia Pacific (Tokyo) Region       | <code>http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Asia Pacific (Seoul) Region       | <code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Asie-Pacifique (Singapour) | <code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |

| Région                             | Emplacement du CloudFormation modèle                                                                                                  |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Région Asia Pacific (Sydney)       | <code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>  |
| Région Asie-Pacifique (Melbourne)  | <code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Région Asie-Pacifique (Mumbai)     | <code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>          |
| Région Amérique du Sud (São Paulo) | <code>aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code>              |

Cette commande crée une CloudFormation pile nommée **CodeDeployDemoStack**, en utilisant le CloudFormation modèle du compartiment Amazon S3 spécifié. L'instance Amazon EC2 est basée sur le type d'instance `t1.micro`, mais vous pouvez utiliser n'importe quel type. Elle est balisée avec la valeur **CodeDeployDemo**, mais vous pouvez la baliser avec une valeur quelconque. La paire de clés d'instance spécifiée lui est appliquée.

2. Appelez la `describe-stacks` commande pour vérifier que la CloudFormation pile nommée **CodeDeployDemoStack** a été créée avec succès :

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query "Stacks[0].StackStatus" --output text
```

Ne continuez pas tant que la valeur `CREATE_COMPLETE` n'est pas retournée.

Pour vérifier que l'agent CodeDeploy s'exécute sur l'instance Amazon EC2, consultez [Gestion des opérations des agents CodeDeploy](#), puis passez à [Créez une application avec CodeDeploy](#).

## Configurer une instance Amazon EC2 pour qu'elle fonctionne avec CodeDeploy

Ces instructions vous montrent comment configurer une instance Amazon EC2 exécutant Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) ou Windows Server pour une utilisation dans le cadre de déploiements. CodeDeploy

### Note

Si vous ne possédez pas d'instance Amazon EC2, vous pouvez utiliser le CloudFormation modèle pour en lancer une exécutant Amazon Linux ou Windows Server. Nous ne fournissons pas de modèle pour Ubuntu Server ou RHEL.

### Étape 1 : vérifier qu'un profil d'instance IAM est attaché à votre instance Amazon EC2

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/ec2/>.
2. Dans le panneau de navigation, sous Instances, choisissez Instances.
3. Accédez à votre instance Amazon EC2 et choisissez-la dans la liste.
4. Dans le volet de détails, sous l'onglet Description, notez la valeur dans le champ du rôle IAM, puis passez à la section suivante.

Si le champ est vide, vous pouvez associer un profil d'instance IAM à l'instance. Pour plus d'informations, consultez [Attacher un rôle IAM à une instance](#).

### Étape 2 : vérifier que le profil d'instance IAM attaché dispose des autorisations d'accès correctes

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles.

3. Accédez au nom du rôle IAM que vous avez noté à l'étape 4 de la section précédente et choisissez-le.

 Note

Si vous souhaitez utiliser le rôle de service généré par le CloudFormation modèle au lieu de celui que vous avez créé en suivant les instructions [Étape 2 : créer un rôle de service pour CodeDeploy](#), notez ce qui suit :

Dans certaines versions de notre CloudFormation modèle, le nom d'affichage du profil d'instance IAM généré et attaché aux instances Amazon EC2 n'est pas le même que le nom d'affichage dans la console IAM. Par exemple, le profil d'instance IAM peut avoir un nom d'affichage de `CodeDeploySampleStack-expny16-InstanceRoleInstanceProfile-IK8J8A9123EX`, tandis que le profil d'instance IAM dans la console IAM peut avoir un nom d'affichage de `CodeDeploySampleStack-expny16-InstanceRole-C5P33V1L64EX`.

Pour vous aider à identifier le profil d'instance dans la console IAM, vous verrez que le préfixe `CodeDeploySampleStack-expny16-InstanceRole` est le même pour les deux. Pour plus d'informations sur les raisons pour lesquelles ces noms d'affichage peuvent être différents, consultez la section [Profils d'instance](#).

4. Choisissez l'onglet Relations d'approbation. Si aucune entrée n'indique The identity provider (s) `ec2.amazonaws.com` dans Trusted Entities, vous ne pouvez pas utiliser cette instance Amazon EC2. Arrêtez et créez une instance Amazon EC2 à l'aide des informations contenues dans [Utilisation d'instances pour CodeDeploy](#).

Si une entrée indique The identity provider (s) `ec2.amazonaws.com` et que vous stockez vos applications uniquement dans des GitHub référentiels, passez directement à [Étape 3 : baliser l'instance Amazon EC2](#).

Si une entrée indique The identity provider (s) `ec2.amazonaws.com` et que vous stockez vos applications dans des compartiments Amazon S3, choisissez l'onglet Autorisations.

5. S'il y a une stratégie dans la zone Managed Policies (Stratégies gérées), choisissez le nom de la stratégie, puis choisissez Edit policy (Modifier la stratégie).
6. Choisissez l'onglet JSON. Si vous stockez vos applications dans des compartiments Amazon S3, assurez-vous qu'"`s3:Get*`" et "`s3:List*`" figurent dans la liste des actions spécifiées.

Cela peut ressembler à ce qui suit :



```
{"Statement":[{"Resource":"*","Action":[
 ... Some actions may already be listed here ...
 "s3:Get*", "s3:List*"
 ... Some more actions may already be listed here ...
], "Effect": "Allow"}]}
```

Ou, cela peut ressembler à ce qui suit :

JSON

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

Si "s3:Get\*" et "s3:List\*" ne sont pas dans la liste des actions spécifiées, choisissez Modifier pour les ajouter, puis choisissez Enregistrer. (Si ni "s3:Get\*", ni "s3:List\*" n'est la dernière action de la liste, veuillez à ajouter une virgule après l'action afin que le document de stratégie soit validé.)

#### Note

Nous vous recommandons de limiter cette politique aux seuls compartiments Amazon S3 auxquels vos instances Amazon EC2 doivent accéder. Assurez-vous de donner accès aux compartiments Amazon S3 contenant l' CodeDeploy agent. Dans le cas contraire, une erreur peut se produire lors de l'installation ou de la mise à jour de l' CodeDeploy agent sur les instances. Pour accorder au profil d'instance IAM l'accès à certains compartiments du kit de CodeDeploy ressources uniquement dans Amazon S3,

appliquez la politique suivante, mais supprimez les lignes relatives aux compartiments auxquels vous souhaitez empêcher l'accès :

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",
 "arn:aws:s3:::aws-codedeploy-us-east-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-2/*",
 "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
 "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
 "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
 "arn:aws:s3:::aws-codedeploy-il-central-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
 "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
 "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
 "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
 "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
 "arn:aws:s3:::aws-codedeploy-me-central-1/*",
 "arn:aws:s3:::aws-codedeploy-me-south-1/*",
 "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
]
 }
]
}
```

```
]
}
```

### Étape 3 : baliser l'instance Amazon EC2

Pour obtenir des instructions sur la façon de baliser l'instance Amazon EC2 afin qu'elle CodeDeploy puisse être détectée lors d'un déploiement, consultez la section [Utilisation des balises dans la console](#), puis revenez à cette page.

#### Note

Vous pouvez étiqueter l'instance Amazon EC2 avec la clé et la valeur de votre choix. Assurez-vous juste de spécifier cette clé et cette valeur lors du déploiement.

### Étape 4 : Installation de l' AWS CodeDeploy agent sur l'instance Amazon EC2

Pour obtenir des instructions sur la façon d'installer l' CodeDeploy agent sur l'instance Amazon EC2 et de vérifier son exécution, consultez [Gestion des opérations des CodeDeploy agents](#), puis passez à [Créez une application avec CodeDeploy](#)

## Utilisation d'instances locales pour CodeDeploy

Une instance sur site est un appareil physique autre qu'une instance Amazon EC2 capable d'exécuter l'agent et de se connecter à CodeDeploy des points de terminaison de service AWS public.

Le déploiement d'une révision d' CodeDeploy application sur une instance locale implique deux étapes principales :

- Étape 1 — Configurez chaque instance locale, enregistrez-la auprès de celle-ci CodeDeploy, puis étiquetez-la.
- Étape 2 — Déployez les révisions de l'application sur l'instance locale.

#### Note

Pour tester la création et le déploiement d'un exemple de révision d'application sur une instance sur site correctement configurée et inscrite, consultez la page [Tutoriel : Déployer](#)

[une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#). Pour plus d'informations sur les instances locales et leur mode de fonctionnement CodeDeploy, consultez [Working with On-Premises Instances](#).

Si vous ne souhaitez plus qu'une instance locale soit utilisée dans les déploiements, vous pouvez supprimer les balises d'instance locales des groupes de déploiement. Pour une approche plus robuste, supprimez les balises d'instance sur site de l'instance. Vous pouvez également annuler explicitement l'inscription d'une instance sur site pour qu'elle ne puisse plus être utilisée dans aucun déploiement. Pour de plus amples informations, veuillez consulter [Gestion des opérations des instances sur site dans CodeDeploy](#).

Les instructions de cette section vous montrent comment configurer une instance sur site, puis l'enregistrer et l'étiqueter afin qu'elle CodeDeploy puisse être utilisée dans des déploiements. Cette section décrit également comment CodeDeploy obtenir des informations sur les instances locales et annuler l'enregistrement d'une instance locale une fois que vous n'avez plus l'intention de la déployer.

## Rubriques

- [Conditions préalables à la configuration d'une instance sur site](#)
- [Enregistrez une instance sur site auprès de CodeDeploy](#)
- [Gestion des opérations des instances sur site dans CodeDeploy](#)

## Conditions préalables à la configuration d'une instance sur site

Les conditions préalables suivantes doivent être satisfaites avant que vous puissiez enregistrer une instance sur site.

### Important

Si vous utilisez la [register-on-premises-instance](#) commande et que vous actualisez régulièrement les informations d'identification temporaires générées avec le AWS Security Token Service (AWS STS), il existe d'autres conditions préalables. Pour plus d'informations, consultez [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#).

## Exigences relatives aux dispositifs

L'appareil que vous souhaitez préparer, enregistrer et étiqueter en tant qu'instance locale CodeDeploy doit exécuter un système d'exploitation compatible. Pour en obtenir la liste, consultez [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#).

Si votre système d'exploitation n'est pas pris en charge, l' CodeDeploy agent est disponible en open source pour que vous puissiez l'adapter à vos besoins. Pour plus d'informations, consultez le référentiel d'[CodeDeploy agents](#) dans GitHub.

## Communications sortantes

L'instance locale doit être en mesure de se connecter aux points de terminaison du AWS service public avec lesquels communiquer. CodeDeploy

L' CodeDeploy agent communique avec le trafic sortant via le protocole HTTPS via le port 443.

## Contrôle administratif

Le compte local ou réseau utilisé sur l'instance locale pour configurer l'instance locale doit pouvoir s'exécuter en tant que sudo ou root (pour Ubuntu Server) ou en tant qu'administrateur (pour Windows Server).

## Autorisations IAM

L'identité IAM que vous utilisez pour enregistrer l'instance locale doit être autorisée à terminer l'enregistrement (et à annuler l'enregistrement de l'instance locale, le cas échéant).

Outre la politique décrite dans [Étape 3 : Limiter les autorisations de l' CodeDeploy utilisateur](#), assurez-vous que l'identité IAM appelante est associée à la politique supplémentaire suivante.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:CreateAccessKey",
 "iam:CreateUser",
 "iam>DeleteAccessKey",
 "iam>DeleteUser",
 "iam>DeleteUserPolicy",
```

```
 "iam:ListAccessKeys",
 "iam:ListUserPolicies",
 "iam:PutUserPolicy",
 "iam:GetUser"
],
 "Resource": "*"
 }
]
```

Pour plus d'informations sur la façon de joindre des stratégies IAM, veuillez consulter [Gestion des stratégies IAM](#).

## Enregistrez une instance sur site auprès de CodeDeploy

Pour enregistrer une instance sur site, vous devez utiliser une identité IAM pour authentifier vos requêtes. Vous pouvez choisir parmi les options suivantes l'identité IAM et la méthode d'enregistrement à utiliser :

- Utiliser un ARN de rôle IAM pour authentifier les requêtes.
- Utilisez la [register-on-premises-instance](#) commande et les informations d'identification temporaires régulièrement actualisées générées avec le AWS Security Token Service (AWS STS) pour configurer manuellement la plupart des options d'enregistrement. Cette option offre le plus haut niveau de sécurité, car l'authentification s'effectue à l'aide d'un jeton temporaire expirant et devant être actualisé régulièrement. Cette option est recommandée pour les déploiements de production de toute taille. Pour plus d'informations, consultez [Utilisez la register-on-premises-instance commande \(ARN de session IAM\) pour enregistrer une instance sur site](#).
- (Non recommandé) Utilisez l'ARN d'un utilisateur IAM pour authentifier les demandes.
  - Utilisez la commande [register](#) pour le processus d'enregistrement le plus automatisé possible. Cette option ne doit être utilisée que pour les déploiements hors production où la sécurité est moins préoccupante. Cette option est moins sécurisée car elle utilise des informations d'identification statiques (permanentes) pour l'authentification. Cette option fonctionne bien pour enregistrer une seule instance sur site. Pour plus d'informations, consultez [Utilisez la commande register \(ARN de l'utilisateur IAM\) pour enregistrer une instance sur site](#).
  - Utilisez la commande [register-on-premises-instance](#) pour configurer manuellement la plupart des options d'enregistrement. Cette méthode convient pour l'enregistrement d'un petit nombre

d'instances locales. Pour plus d'informations, consultez [Utilisez la register-on-premises-instance commande \(ARN de l'utilisateur IAM\) pour enregistrer une instance sur site](#).

## Rubriques

- [Utilisez la register-on-premises-instance commande \(ARN de session IAM\) pour enregistrer une instance sur site](#)
- [Utilisez la commande register \(ARN de l'utilisateur IAM\) pour enregistrer une instance sur site](#)
- [Utilisez la register-on-premises-instance commande \(ARN de l'utilisateur IAM\) pour enregistrer une instance sur site](#)

Utilisez la register-on-premises-instance commande (ARN de session IAM) pour enregistrer une instance sur site

Pour un contrôle maximal de l'authentification et de l'enregistrement de vos instances locales, vous pouvez utiliser la [register-on-premises-instance](#) commande et actualiser régulièrement les informations d'identification temporaires générées avec le AWS Security Token Service (AWS STS). Un rôle IAM statique pour l'instance assume le rôle de ces informations d' AWS STS identification actualisées pour effectuer les opérations de CodeDeploy déploiement.

Cette méthode est très utile lorsque vous devez enregistrer un grand nombre d'instances. Il vous permet d'automatiser le processus d'inscription avec CodeDeploy. Vous pouvez utiliser votre propre système d'identité et d'authentification pour authentifier les instances sur site et distribuer les informations d'identification de session IAM du service aux instances à utiliser avec. CodeDeploy

### Note

Vous pouvez également utiliser un utilisateur IAM partagé distribué à toutes les instances locales pour appeler l' AWS STS [AssumeRole](#) API afin de récupérer les informations d'identification de session pour les instances locales. Cette méthode est moins sécurisée et n'est pas recommandée pour les environnements critiques ou de production.

Utilisez les informations des rubriques suivantes pour configurer une instance sur site à l'aide des informations d'identification de sécurité temporaires générées avec AWS STS.

## Rubriques

- [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#)
- [Étape 1 : créer le rôle IAM que les instances locales assumeront](#)
- [Étape 2 : générer des informations d'identification temporaires pour une instance individuelle à l'aide de AWS STS](#)
- [Étape 3 : ajouter un fichier de configuration à l'instance locale](#)
- [Étape 4 : préparer une instance sur site pour CodeDeploy les déploiements](#)
- [Étape 5 : enregistrer l'instance sur site auprès de CodeDeploy](#)
- [Étape 6 : baliser l'instance locale](#)
- [Étape 7 : Déployer les révisions de l'application sur l'instance locale](#)
- [Étape 8 : suivre les déploiements vers l'instance sur site](#)

## Conditions requises pour l'enregistrement de l'ARN de la session IAM

Outre les prérequis figurant dans [Conditions préalables à la configuration d'une instance sur site](#), les exigences suivantes doivent être respectées :

### Autorisations IAM

L'identité IAM que vous utilisez pour enregistrer une instance sur site doit être autorisée à effectuer CodeDeploy des opérations. Assurez-vous que la stratégie gérée `AWSCodeDeployFullAccess` est associée à l'identité IAM. Pour plus d'informations, consultez [les politiques AWS gérées](#) dans le guide de l'utilisateur IAM.

### Système de renouvellement des informations d'identification temporaires

Si vous utilisez un ARN de session IAM pour enregistrer des instances sur site, il vous faut mettre en place un système pour actualiser régulièrement les informations d'identification temporaires. Les informations d'identification temporaires expirent après une heure, voire plus tôt si vous spécifiez un délai plus court lorsque les informations d'identification sont générées. Il existe deux méthodes pour actualiser les informations d'identification :

- Méthode 1 : utilisez le système d'identité et d'authentification en place au sein de votre réseau d'entreprise avec un script CRON qui interroge régulièrement le système d'authentification et d'identité afin de copier les dernières informations d'identification de session vers l'instance. Cela vous permet d'intégrer votre authentification et votre structure d'identité AWS sans avoir à modifier l' CodeDeploy agent ou le service pour prendre en charge les types d'authentification que vous utilisez dans votre organisation.



- Méthode 2 : exécutez régulièrement une tâche CRON sur l'instance pour appeler l' [AWS STS AssumeRole](#) action et écrire les informations d'identification de session dans un fichier auquel l' CodeDeploy agent peut accéder. Cette méthode nécessite d'un utilisateur IAM et la copie des informations d'identification sur l'instance sur site, mais vous pouvez réutiliser le même utilisateur IAM et les informations d'identification pour l'ensemble de votre parc d'instances sur site.

 Note

Que vous utilisiez la méthode 1 ou 2, vous devez configurer un processus pour redémarrer l' CodeDeploy agent après la mise à jour des informations d'identification de session temporaires afin que les nouvelles informations d'identification prennent effet.

Pour plus d'informations sur la création et l'utilisation des AWS STS informations d'identification, consultez les [AWS Security Token Service sections Référence d'API](#) et [Utilisation d'informations d'identification de sécurité temporaires pour demander l'accès aux AWS ressources](#).

Étape 1 : créer le rôle IAM que les instances locales assumeront

Vous pouvez utiliser la console AWS CLI ou la console IAM pour créer un rôle IAM qui sera utilisé par vos instances locales pour s'authentifier et interagir avec. CodeDeploy

Vous n'avez qu'un seul rôle IAM à créer. Chacune de vos instances sur site peut endosser ce rôle pour récupérer les informations d'identification de sécurité temporaires et obtenir les autorisations accordées à ce rôle.

Le rôle que vous créez nécessite les autorisations suivantes pour accéder aux fichiers nécessaires à l'installation de l' CodeDeploy agent :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],

```

```
 "Effect": "Allow",
 "Resource": "*"
 }
]
```

Nous vous recommandons de limiter cette politique aux seuls compartiments Amazon S3 auxquels votre instance sur site doit accéder. Si vous limitez cette politique, assurez-vous de donner accès aux compartiments Amazon S3 contenant l' CodeDeploy agent. Dans le cas contraire, une erreur peut se produire chaque fois que l' CodeDeploy agent est installé ou mis à jour sur l'instance locale. Pour plus d'informations sur le contrôle de l'accès aux compartiments Amazon S3, consultez [Gestion des autorisations d'accès à vos ressources Amazon S3](#).

Pour créer le rôle IAM

1. Appelez la commande [create-role](#) avec l'option `--role-name` afin de préciser le nom du rôle IAM (par exemple, `CodeDeployInstanceRole`) et l'option `--assume-role-policy-document` pour fournir les autorisations.

Lorsque vous créez le rôle IAM pour cette instance, vous pouvez lui attribuer le nom `CodeDeployInstanceRole` et inclure les autorisations requises dans un fichier nommé `CodeDeployRolePolicy.json` :

```
aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-
document file://CodeDeployRolePolicy.json
```

2. Dans la sortie de l'appel de la commande `create-role`, notez la valeur du champ ARN. Par exemple :

```
arn:aws:iam::123456789012:role/CodeDeployInstanceRole
```

Vous aurez besoin de l'ARN du rôle lorsque vous utiliserez l' AWS STS [AssumeRole](#) API pour générer des informations d'identification à court terme pour chaque instance.

Pour plus d'informations sur la création de rôles IAM, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur l'attribution d'autorisations à un rôle existant, consultez la section [Référence put-role-policy des AWS CLI commandes](#).

## Étape 2 : générer des informations d'identification temporaires pour une instance individuelle à l'aide de AWS STS

Avant de générer les informations d'identification temporaires qui serviront à enregistrer l'instance sur site, vous devez créer ou choisir l'identité IAM (utilisateur ou rôle) pour laquelle vous allez générer les informations d'identification temporaires. L'autorisation `sts:AssumeRole` doit être incluse dans les paramètres de stratégie de cette identité IAM.

Pour plus d'informations sur l'octroi d'`sts:AssumeRole` autorisations à une identité IAM, consultez [Création d'un rôle pour déléguer des autorisations à un AWS service](#) et [AssumeRole](#).

Il existe deux façons de générer les informations d'identification temporaires :

- Utilisez la commande [assume-role](#) avec AWS CLI. Par exemple :

```
aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-session-name session-name
```

Où :

- 12345ACCOUNT* est le numéro de compte à 12 chiffres de votre organisation.
- role-arn* est l'ARN du rôle à assumer, que vous avez généré dans [Étape 1 : créer le rôle IAM que les instances locales assumeront](#).
- session-name* est le nom que vous souhaitez attribuer à la session de rôle que vous êtes en train de créer.

### Note

Si vous utilisez un script CRON qui interroge régulièrement le système d'identité et d'authentification et copie les dernières informations d'identification de session sur l'instance (méthode 1 pour actualiser les informations d'identification temporaires décrite dans [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#)), vous pouvez utiliser n'importe quel AWS SDK compatible pour appeler [AssumeRole](#).

- Utilisez un outil fourni par AWS.

L' `aws-codedeploy-session-helper` outil génère des AWS STS informations d'identification et les écrit dans un fichier que vous placez sur l'instance. Cet outil est mieux adapté à la méthode 2 d'actualisation des informations d'identification temporaires, décrite dans [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#). Dans cette méthode, l' `aws-codedeploy-session-`

helperoutil est placé sur chaque instance et exécute la commande en utilisant les autorisations d'un utilisateur IAM. Chaque instance utilise cet outil avec les informations d'identification du même utilisateur IAM.

Pour plus d'informations, consultez le [aws-codedeploy-session-helper](#) GitHub référentiel.

 Note

Une fois que vous avez créé les informations d'identification de la session IAM, placez-les à n'importe quel emplacement sur l'instance sur site. À l'étape suivante, vous allez configurer l' CodeDeploy agent pour accéder aux informations d'identification à cet emplacement.

Avant de poursuivre, assurez-vous que le système que vous allez utiliser pour actualiser régulièrement les informations d'identification temporaires est en place. Si les informations d'identification temporaires ne sont pas actualisées, les déploiements vers l'instance sur site échoueront. Pour plus d'informations, consultez « Système de renouvellement des informations d'identification temporaires » dans [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#).

### Étape 3 : ajouter un fichier de configuration à l'instance locale

Ajoutez un fichier de configuration à l'instance sur site en utilisant les autorisations racine ou d'administrateur. Ce fichier de configuration est utilisé pour déclarer les informations d'identification IAM et la AWS région cible à CodeDeploy utiliser. Le fichier doit être ajouté à un emplacement spécifique sur l'instance sur site. Le fichier doit inclure l'ARN de la session temporaire IAM, son identifiant de clé secrète et sa clé d'accès secrète, ainsi que la AWS région cible.

Pour ajouter un fichier de configuration

1. Créez un fichier nommé `codedeploy.onpremises.yml` (pour un serveur Ubuntu ou une instance locale RHEL) ou `conf.onpremises.yml` (pour une instance locale Windows Server) à l'emplacement suivant sur l'instance locale :
  - Pour le serveur Ubuntu : `/etc/codedeploy-agent/conf`
  - Pour Windows Server : `C:\ProgramData\Amazon\CodeDeploy`

2. Utilisez un éditeur de texte pour ajouter les informations suivantes au `codedeploy.onpremises.yml` fichier (Linux) ou au `conf.onpremises.yml` fichier (Windows) nouvellement créé :

```

iam_session_arn: iam-session-arn
aws_credentials_file: credentials-file
region: supported-region
```

Où :

- *iam-session-arn* est l'ARN de session IAM dans [Étape 2 : générer des informations d'identification temporaires pour une instance individuelle à l'aide de AWS STS](#) lequel vous l'avez noté.
- *credentials-file* est l'emplacement du fichier d'informations d'identification pour l'ARN de session temporaire, comme indiqué dans [Étape 2 : générer des informations d'identification temporaires pour une instance individuelle à l'aide de AWS STS](#).
- *supported-region* est l'une des régions que CodeDeploy prend en charge, comme indiqué dans [Région et points de terminaison](#) dans Références générales AWS.

## Étape 4 : préparer une instance sur site pour CodeDeploy les déploiements

### Installez et configurez AWS CLI

Installez et configurez l'instance AWS CLI sur site. (Le AWS CLI sera utilisé pour télécharger et installer l' CodeDeploy agent sur l'instance locale.)

1. Pour installer l'instance AWS CLI sur site, suivez les instructions de la [section Configuration avec le guide AWS CLI de l'AWS Command Line Interface utilisateur](#).

#### Note

CodeDeploy les commandes permettant de travailler avec des instances locales sont devenues disponibles dans la version 1.7.19 du AWS CLI. Si vous avez une version AWS CLI déjà installée, vous pouvez vérifier sa version en appelant `aws --version`.

2. Pour configurer l'instance AWS CLI sur site, suivez les instructions de la [section Configuration de l'instance AWS CLI dans le guide de l'AWS Command Line Interface utilisateur](#).

 Important

Lorsque vous configurez le AWS CLI (par exemple, en appelant la `aws configure` commande), veillez à spécifier l'ID de clé secrète et la clé d'accès secrète d'un utilisateur IAM disposant au minimum des autorisations décrites dans [Conditions requises pour l'enregistrement de l'ARN de la session IAM](#).

Définissez la variable d' `AWS_REGION` d'environnement (Ubuntu Server et RHEL uniquement)

Si vous n'exécutez pas Ubuntu Server ou RHEL sur votre instance locale, ignorez cette étape et passez directement à « Installer l' CodeDeploy agent ».

Installez l' CodeDeploy agent sur un serveur Ubuntu ou une instance RHEL sur site et autorisez l'instance à mettre à jour l' CodeDeploy agent chaque fois qu'une nouvelle version est disponible. Pour ce faire, définissez la variable d'environnement `AWS_REGION` sur l'instance en spécifiant l'identifiant d'une des régions prises en charge par CodeDeploy. Nous vous recommandons de définir la valeur en fonction de la région dans laquelle se trouvent vos CodeDeploy applications, groupes de déploiement et révisions d'applications (par exemple, `us-west-2`). Pour obtenir la liste des régions, voir [Région et points de terminaison](#) dans le Références générales AWS.

Pour définir la variable d'environnement, appelez la commande suivante depuis le terminal :

```
export AWS_REGION=supported-region
```

Où se *supported-region* trouve l'identifiant de région (par exemple, `us-west-2`).

### Installation de l' CodeDeploy agent

- Pour une instance locale d'Ubuntu Server, suivez les instructions fournies [Installation de l' CodeDeploy agent pour Ubuntu Server](#), puis revenez à cette page.
- Pour une instance RHEL sur site, suivez les instructions fournies [Installation de l' CodeDeploy agent pour Amazon Linux ou RHEL](#), puis revenez à cette page.
- Pour une instance locale de Windows Server, suivez les instructions indiquées dans [Installation de l' CodeDeploy agent pour Windows Server](#), puis revenez à cette page.

## Étape 5 : enregistrer l'instance sur site auprès de CodeDeploy

Les instructions de cette étape supposent que vous inscrivez l'instance sur site à partir de l'instance sur site elle-même. Vous pouvez enregistrer une instance sur site à partir d'un appareil ou d'une instance distinct sur lequel elle est AWS CLI installée et configurée.

Utilisez le AWS CLI pour enregistrer l'instance sur site CodeDeploy afin qu'elle puisse être utilisée dans des déploiements.

Avant de pouvoir utiliser le AWS CLI, vous aurez besoin de l'ARN des informations d'identification de session temporaires que vous avez créées dans [Étape 3 : ajouter un fichier de configuration à l'instance locale](#). Par exemple, pour une instance que vous identifiez comme AssetTag12010298EX :

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Appelez la commande [register-on-premises-instance](#), en spécifiant les éléments suivants :

- Un nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-name`).

### Important

Pour identifier plus facilement l'instance sur site, notamment à des fins de débogage, nous vous recommandons vivement de spécifier un nom faisant référence à une caractéristique unique de l'instance sur site (par exemple, le nom de session des informations d'identification STS et le numéro de série ou un identificateur de ressource interne unique, le cas échéant). Si vous spécifiez une adresse MAC comme nom, sachez que les adresses MAC contiennent des caractères CodeDeploy interdits, tels que deux points (:)). Pour obtenir la liste des caractères autorisés, consultez la page [CodeDeploy quotas](#).

- L'ARN de session IAM que vous avez configuré pour authentifier plusieurs instances sur site à [l'Étape 1 : créer le rôle IAM que les instances locales assumeront](#).

Par exemple :

```
aws deploy register-on-premises-instance --instance-name name-of-instance --iam-session-arn arn:aws:sts::account-id:assumed-role/role-to-assume/session-name
```

Où :

- ***name-of-instance*** est le nom que vous utilisez pour identifier l'instance locale, par exemple. `AssetTag12010298EX`
- ***account-id*** est l'identifiant de compte à 12 chiffres de votre organisation, par exemple `111222333444`.
- ***role-to-assume*** est le nom du rôle IAM que vous avez créé pour l'instance, par exemple. `CodeDeployInstanceRole`
- ***session-name*** est le nom du rôle de session que vous avez spécifié dans [Étape 2 : générer des informations d'identification temporaires pour une instance individuelle à l'aide de AWS STS](#).

## Étape 6 : baliser l'instance locale

Vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour étiqueter l'instance locale. (CodeDeploy utilise des balises d'instance locales pour identifier les cibles de déploiement lors d'un déploiement.)

Pour baliser l'instance sur site (interface CLI)

- Appelez la commande [add-tags-to-on-premises-instances](#) en spécifiant :
  - Le nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-names`).
  - Le nom de la clé de balise et de la valeur de balise de l'instance sur site que vous voulez utiliser (avec l'option `--tags`). Vous devez spécifier à la fois un nom et une valeur. CodeDeploy n'autorise pas les balises d'instance locales contenant uniquement des valeurs.

Par exemple :

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
--tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Pour baliser l'instance sur site (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.



 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez On-premises instances.
3. Dans la liste des instances sur site, choisissez nom de l'instance sur site à baliser.
4. Dans la liste des balises, sélectionnez ou entrez la clé de balise et la valeur de balise souhaitées. Une fois que vous avez entré la clé de balise et la valeur de balise, une autre ligne apparaît. Vous pouvez répéter cela pour un maximum de 10 balises. Pour supprimer une identification, choisissez Supprimer.
5. Après avoir ajouté les balises, choisissez Update Tags.

## Étape 7 : Déployer les révisions de l'application sur l'instance locale

Vous êtes maintenant prêt à déployer les révisions d'application sur l'instance sur site inscrite et balisée.

Vous déployez des révisions d'applications sur des instances sur site de la même manière que le déploiement de révisions d'applications sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Créez un déploiement avec CodeDeploy](#). Ces instructions incluent un lien vers des prérequis, y compris la création d'une application, la création d'un groupe de déploiement et la préparation d'une révision d'application. Si vous avez besoin d'un exemple simple de révision d'application à déployer, vous pouvez créer celui décrit dans [Étape 2 : Création d'un exemple de révision d'application](#), dans la rubrique [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#).

 Important

Si vous réutilisez un rôle de CodeDeploy service dans le cadre de la création d'un groupe de déploiement qui cible les instances locales, vous Tag : get \* devez inclure dans la Action partie de la déclaration de politique du rôle de service. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

## Étape 8 : suivre les déploiements vers l'instance sur site

Après avoir déployé une révision d'application sur les instances sur site inscrites et balisées, vous pouvez suivre la progression du déploiement.

Vous suivez les déploiements sur des instances sur site de la même manière que le suivi des déploiements sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Afficher les détails CodeDeploy du déploiement](#).

Utilisez la commande register (ARN de l'utilisateur IAM) pour enregistrer une instance sur site

### Important

L'enregistrement d'une instance à l'aide d'un utilisateur IAM n'est pas recommandé car il utilise des informations d'identification statiques (permanentes) pour l'authentification. Pour améliorer la sécurité, nous vous recommandons d'enregistrer une instance à l'aide d'informations d'identification temporaires pour l'authentification. Pour de plus amples informations, veuillez consulter [Utilisez la register-on-premises-instance commande \(ARN de session IAM\) pour enregistrer une instance sur site](#).

### Important

Assurez-vous d'avoir mis en place un plan pour alterner les clés d'accès des utilisateurs IAM (informations d'identification permanentes). Pour plus d'informations, voir [Rotation des touches d'accès](#).

Cette section décrit comment configurer une instance sur site, l'enregistrer et la baliser auprès d'CodeDeploy avec un minimum d'efforts. La commande register est très utile lorsque vous travaillez avec une seule instance sur site ou des parcs d'instances locales restreints. Vous ne pouvez utiliser la register commande que lorsque vous utilisez un ARN utilisateur IAM pour authentifier une instance. Vous ne pouvez pas utiliser la register commande avec un ARN de session IAM pour l'authentification.

Lorsque vous utilisez la register commande, vous pouvez CodeDeploy effectuer les opérations suivantes :

- Créez un utilisateur IAM Gestion des identités et des accès AWS pour l'instance sur site, si vous n'en spécifiez aucun avec la commande.
- Enregistrez les informations d'identification de l'utilisateur IAM dans un fichier de configuration d'instance sur site.
- Enregistrez l'instance locale auprès CodeDeploy de.
- Ajouter des balises à l'instance sur site, dès lors que vous les spécifiez dans la commande.

#### Note

La [register-on-premises-instance](#) commande est une alternative à la commande [register](#). Vous utilisez la commande `register-on-premises-instance` si vous souhaitez principalement par vous-même configurer une instance sur site, l'enregistrer et la baliser avec CodeDeploy. La `register-on-premises-instance` commande vous donne également la possibilité d'utiliser un ARN de session IAM pour enregistrer des instances au lieu d'un ARN d'utilisateur IAM. Cette approche constitue un avantage non négligeable si vous avez des parcs importants d'instances locales. Plus précisément, vous pouvez utiliser un seul ARN de session IAM pour authentifier plusieurs instances au lieu de devoir créer un utilisateur IAM pour chaque instance locale une par une. Pour plus d'informations, consultez [Utilisez la register-on-premises-instance commande \(ARN de l'utilisateur IAM\) pour enregistrer une instance sur site](#) et [Utilisez la register-on-premises-instance commande \(ARN de session IAM\) pour enregistrer une instance sur site](#).

## Rubriques

- [Étape 1 : Installation et configuration de l'instance AWS CLI sur site](#)
- [Étape 2 : Appelez la commande d'enregistrement](#)
- [Étape 3 : appelez la commande d'installation](#)
- [Étape 4 : Déployer les révisions de l'application sur l'instance locale](#)
- [Étape 5 : suivre les déploiements vers l'instance sur site](#)

### Étape 1 : Installation et configuration de l'instance AWS CLI sur site

1. Installez l'instance AWS CLI sur site. Suivez les instructions de [la section Configuration avec le guide AWS CLI de l'AWS Command Line Interface utilisateur](#).

 Note

CodeDeploy les commandes permettant de travailler avec des instances locales sont disponibles dans les AWS CLI versions 1.7.19 et ultérieures. Si vous l'avez AWS CLI déjà installé, appelez `aws --version` pour vérifier sa version.

2. Configurez l'AWS CLI instance sur site. Suivez les instructions de la [section Configuration du guide AWS CLI de AWS Command Line Interface l'utilisateur intégré](#).

 Important

Lorsque vous configurez le AWS CLI (par exemple, en appelant la `aws configure` commande), veillez à spécifier l'ID de clé secrète et la clé d'accès secrète d'un utilisateur IAM qui dispose, au minimum, des autorisations AWS d'accès suivantes en plus des autorisations spécifiées dans [Conditions préalables à la configuration d'une instance sur site](#). Cela permet de télécharger et d'installer l'CodeDeploy agent sur l'instance locale. Les autorisations d'accès peuvent ressembler à ceci :

JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*",
 "iam:CreateAccessKey",
 "iam:CreateUser",
 "iam>DeleteAccessKey",
 "iam>DeleteUser",
 "iam>DeleteUserPolicy",
 "iam:ListAccessKeys",
 "iam:ListUserPolicies",
 "iam:PutUserPolicy",
 "iam:GetUser",
 "tag:getTagKeys",
 "tag:getTagValues",
 "tag:GetResources"
]
 }
]
}
```

```

],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "arn:aws:s3:::amzn-s3-demo-bucket1/*"
]
 }
]
}

```

#### Note

Si vous voyez des erreurs d'accès refusé lorsque vous tentez d'accéder à l'un des compartiments Amazon S3 présentés précédemment, essayez d'omettre la /\* partie de l'ARN des ressources du compartiment, par exemple, `arn:aws:s3:::aws-codedeploy-sa-east-1`

## JSON

```

{
 "Version":"2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*",
 "iam:CreateAccessKey",
 "iam:CreateUser",
 "iam>DeleteAccessKey",
 "iam>DeleteUser",
 "iam>DeleteUserPolicy",
 "iam:ListAccessKeys",

```

```
 "iam:ListUserPolicies",
 "iam:PutUserPolicy",
 "iam:GetUser",
 "tag:GetResources"
],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "arn:aws:s3:::amzn-s3-demo-bucket2/*"
]
 }
]
```

## Étape 2 : Appelez la commande d'enregistrement

Pour cette étape, nous supposons que vous inscrivez l'instance sur site à partir de l'instance sur site elle-même. Vous pouvez également enregistrer une instance sur site à partir d'un appareil ou d'une instance distinct sur lequel elle est AWS CLI installée et configurée comme décrit à l'étape précédente.

Utilisez la commande AWS CLI pour appeler le [registre](#), en spécifiant :

- Nom qui identifie de manière unique l'instance locale à CodeDeploy (avec l'option `--instance-name`).

### Important

Pour mieux identifier l'instance sur site ultérieurement, notamment à des fins de débogage, nous vous recommandons vivement d'utiliser un nom mappé à une caractéristique unique de l'instance sur site (par exemple, le numéro de série ou un identificateur de ressource interne unique, le cas échéant). Si vous spécifiez une adresse MAC pour un nom, sachez que les adresses MAC contiennent des CodeDeploy caractères interdits, tels que deux

points (:). Pour obtenir la liste des caractères autorisés, consultez la page [CodeDeploy quotas](#).

- Facultativement, l'ARN d'un utilisateur IAM existant que vous souhaitez associer à cette instance sur site (avec l'option `--iam-user-arn`). Pour obtenir l'ARN d'un utilisateur IAM, appelez la commande [get-user ou choisissez le nom d'utilisateur](#) IAM dans la section Utilisateurs de la console IAM, puis recherchez la valeur de l'ARN utilisateur dans la section Résumé. Si cette option n'est pas spécifiée, un utilisateur IAM CodeDeploy sera créé en votre nom dans votre AWS compte et l'associera à l'instance locale.

 Important

Si vous spécifiez l'option `--iam-user-arn`, vous devez également créer manuellement le fichier de configuration de l'instance sur site, comme décrit dans [Étape 4 : ajouter un fichier de configuration à l'instance locale](#).

Vous ne pouvez associer qu'un seul utilisateur IAM à une seule instance locale. Essayer d'associer un seul utilisateur IAM à plusieurs instances locales peut entraîner des erreurs, des échecs de déploiements sur ces instances sur site ou des déploiements sur des instances sur site bloquées dans un état d'attente perpétuel.

- Facultativement, un ensemble de balises d'instance sur site (avec l'option `--tags`) qui CodeDeploy seront utilisées pour identifier l'ensemble d'instances Amazon EC2 sur lesquelles effectuer le déploiement. Spécifiez chaque balise avec `Key=tag-key, Value=tag-value` (par exemple, `Key=Name, Value=Beta` `Key=Name, Value=WestRegion`). Si cette option n'est pas spécifiée, aucune balise ne sera inscrite. Pour enregistrer les balises ultérieurement, appelez la commande [add-tags-to-on-premises-instances](#).
- Facultativement, la AWS région dans laquelle l'instance locale sera enregistrée CodeDeploy (avec l'option `--region`). Il doit s'agir de l'une des régions prises en charge répertoriées dans [Région et des points de terminaison](#) dans Références générales AWS (par exemple, `us-west-2`). Si cette option n'est pas spécifiée, la AWS région par défaut associée à l'utilisateur IAM appelant sera utilisée.

Par exemple :

```
aws deploy register --instance-name AssetTag12010298EX --iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem --tags Key=Name,Value=CodeDeployDemo-OnPrem --region us-west-2
```

La commande `register` exécute les opérations suivantes :

1. Si aucun utilisateur IAM existant n'est spécifié, crée un utilisateur IAM, lui attache les autorisations requises et génère une clé secrète et une clé d'accès secrète correspondantes. L'instance sur site utilisera cet utilisateur IAM ainsi que ses autorisations et informations d'identification pour s'authentifier et interagir avec lui. CodeDeploy
2. Enregistre l'instance locale auprès CodeDeploy de.
3. Si cela est spécifié, CodeDeploy les balises spécifiées avec l'option `--tag` sont associées au nom de l'instance locale enregistrée.
4. Si un utilisateur IAM a été créé, il crée également le fichier de configuration requis dans le répertoire à partir duquel la `register` commande a été appelée.

Si cette commande rencontre des erreurs, un message d'erreur s'affiche, qui décrit comment suivre manuellement les étapes restantes. Sinon, un message de réussite s'affiche, qui décrit comment appeler la commande `install`, telle qu'elle est répertoriée à l'étape suivante.

Étape 3 : appelez la commande d'installation

À partir de l'instance locale, utilisez la commande AWS CLI pour appeler l'[installation](#), en spécifiant :

- Le chemin d'accès au fichier de configuration (avec l'option `--config-file`).
- Eventuellement, s'il faut remplacer le fichier de configuration qui existe déjà sur l'instance sur site (avec l'option `--override-config`). Si cette option n'est pas spécifiée, le fichier de configuration existant n'est pas remplacé.
- Facultativement, la AWS région dans laquelle l'instance locale sera enregistrée CodeDeploy (avec l'option `--region`). Il doit s'agir de l'une des régions prises en charge répertoriées dans [Région et des points de terminaison](#) dans Références générales AWS(par exemple,us-west-2). Si cette option n'est pas spécifiée, la AWS région par défaut associée à l'utilisateur IAM appelant sera utilisée.
- Facultativement, un emplacement personnalisé à partir duquel installer l' CodeDeploy agent (avec l'option `--agent-install`). Cette option est utile pour installer une version personnalisée de l' CodeDeploy agent qui n'est CodeDeploy pas officiellement prise en charge (telle qu'une version personnalisée basée sur le référentiel de l'[CodeDeployagent](#) dans GitHub). La valeur doit être le chemin d'accès à un compartiment Amazon S3 contenant soit :
  - Script d'installation de l' CodeDeploy agent (pour les systèmes d'exploitation basés sur Linux ou Unix, similaire au fichier d'installation dans le référentiel de l'[CodeDeployagent](#) dans GitHub).



- Un fichier du package d'installation de l' CodeDeploy agent (.msi) (pour les systèmes d'exploitation Windows).

Si cette option n'est pas spécifiée, il CodeDeploy fera de son mieux pour installer depuis son propre emplacement une version officiellement prise en charge de l' CodeDeploy agent compatible avec le système d'exploitation de l'instance locale.

Par exemple :

```
aws deploy install --override-config --config-file /tmp/codedeploy.onpremises.yml --
region us-west-2 --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-
agent.msi
```

La commande `install` exécute les opérations suivantes :

1. Vérifie si l'instance sur site est une instance Amazon EC2. Si c'est le cas, un message d'erreur s'affiche.
2. Copie le fichier de configuration des instances locales de l'emplacement spécifié sur l'instance vers l'emplacement où l' CodeDeploy agent s'attend à le trouver, à condition que le fichier ne se trouve pas déjà à cet emplacement.

Pour Ubuntu Server et Red Hat Enterprise Linux (RHEL), il s'agit de `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`.

Pour Windows Server, il s'agit de `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.

Si l'option `--override-config` a été spécifiée, la commande crée ou remplace le fichier.

3. Installe l' CodeDeploy agent sur l'instance locale, puis le démarre.

Étape 4 : Déployer les révisions de l'application sur l'instance locale

Vous êtes maintenant prêt à déployer les révisions d'application sur l'instance sur site inscrite et balisée.

Vous déployez des révisions d'applications sur des instances sur site de la même manière que le déploiement de révisions d'applications sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Créez un déploiement avec CodeDeploy](#). Ces instructions sont liées

à des prérequis, y compris à la création d'une application, à la création d'un groupe de déploiement et à la préparation d'une révision d'application. Si vous avez besoin d'un exemple simple de révision d'application à déployer, vous pouvez créer celui décrit dans [Étape 2 : Création d'un exemple de révision d'application](#), dans la rubrique [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#).

 Important

Si vous réutilisez un rôle de CodeDeploy service existant dans le cadre de la création d'un groupe de déploiement qui cible les instances locales, vous `Tag: get*` devez inclure dans la `Action` partie de la déclaration de politique du rôle de service. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

## Étape 5 : suivre les déploiements vers l'instance sur site

Après avoir déployé une révision d'application sur les instances sur site inscrites et balisées, vous pouvez suivre la progression du déploiement.

Vous suivez les déploiements sur des instances sur site de la même manière que le suivi des déploiements sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Afficher les détails CodeDeploy du déploiement](#).

Pour découvrir plus d'options, consultez [Gestion des opérations des instances sur site dans CodeDeploy](#).

Utilisez la `register-on-premises-instance` commande (ARN de l'utilisateur IAM) pour enregistrer une instance sur site

 Important

L'enregistrement d'une instance à l'aide d'un utilisateur IAM n'est pas recommandé car il utilise des informations d'identification statiques (permanentes) pour l'authentification. Pour améliorer la sécurité, nous vous recommandons d'enregistrer une instance à l'aide d'informations d'identification temporaires pour l'authentification. Pour de plus amples informations, veuillez consulter [Utilisez la register-on-premises-instance commande \(ARN de session IAM\) pour enregistrer une instance sur site](#).

 Important

Assurez-vous d'avoir mis en place un plan pour alterner les clés d'accès des utilisateurs IAM (informations d'identification permanentes). Pour plus d'informations, voir [Rotation des touches d'accès](#).

Suivez les instructions suivantes pour configurer une instance sur site, l'enregistrer et la baliser avec CodeDeploy de manière principalement manuelle et en utilisant les informations d'identification d'un utilisateur IAM statique pour l'authentification.

## Rubriques

- [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#)
- [Étape 2 : attribuer des autorisations à l'utilisateur IAM](#)
- [Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#)
- [Étape 4 : ajouter un fichier de configuration à l'instance locale](#)
- [Étape 5 : Installation et configuration du AWS CLI](#)
- [Étape 6 : définir la variable d' AWS\\_REGION environnement \(Ubuntu Server et RHEL uniquement\)](#)
- [Étape 7 : Installation de l' CodeDeploy agent](#)
- [Étape 8 : enregistrer l'instance locale auprès de CodeDeploy](#)
- [Étape 9 : baliser l'instance locale](#)
- [Étape 10 : Déployer les révisions de l'application sur l'instance locale](#)
- [Étape 11 : suivre les déploiements vers l'instance sur site](#)

## Étape 1 : créer un utilisateur IAM pour l'instance sur site

Créez un utilisateur IAM que l'instance locale utilisera pour s'authentifier et interagir avec lui.  
CodeDeploy

 Important

Vous devez créer un utilisateur IAM distinct pour chaque instance locale participante. Si vous essayez de réutiliser un utilisateur IAM individuel pour plusieurs instances locales, il se peut que vous ne puissiez pas enregistrer ou étiqueter correctement ces instances sur site.

CodeDeploy Les déploiements sur ces instances sur site risquent d'être bloqués dans un état en attente perpétuel ou d'échouer complètement.

Nous vous recommandons d'attribuer à l'utilisateur IAM un nom identifiant son objectif, tel que CodeDeployUser -OnPrem.

Vous pouvez utiliser la console AWS CLI ou la console IAM pour créer un utilisateur IAM. Pour plus d'informations, consultez [la section Création d'un utilisateur IAM dans votre AWS compte](#).

 Important

Que vous utilisiez la console AWS CLI ou la console IAM pour créer un nouvel utilisateur IAM, notez l'ARN de l'utilisateur fourni à cet utilisateur. Vous aurez besoin de ces informations plus tard dans [Étape 4 : ajouter un fichier de configuration à l'instance locale](#) et [Étape 8 : enregistrer l'instance locale auprès de CodeDeploy](#).

## Étape 2 : attribuer des autorisations à l'utilisateur IAM

Si votre instance sur site doit déployer des révisions d'applications à partir de compartiments Amazon S3, vous devez attribuer à l'utilisateur IAM les autorisations nécessaires pour interagir avec ces compartiments. Vous pouvez utiliser la console AWS CLI ou la console IAM pour attribuer des autorisations.

 Note

Si vous comptez déployer des révisions d'applications uniquement à partir de GitHub référentiels, ignorez cette étape et passez directement à [Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#). (Vous aurez toujours besoin d'informations sur l'utilisateur IAM que vous avez créé dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#). Il sera utilisé dans les étapes ultérieures.)

Pour attribuer des autorisations (interface CLI)

1. Créez un fichier avec le contenu de politique suivant sur l'instance ou l'appareil Amazon EC2 que vous utilisez pour appeler le. AWS CLI Attribuez au fichier un nom comme **CodeDeploy-OnPrem-Permissions.json**, puis enregistrez le fichier.

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

#### Note

Nous vous recommandons de limiter cette politique aux seuls compartiments Amazon S3 auxquels votre instance sur site doit accéder. Si vous limitez cette politique, assurez-vous de donner également accès aux compartiments Amazon S3 contenant l' AWS CodeDeploy agent. Dans le cas contraire, une erreur peut se produire chaque fois que l' CodeDeploy agent est installé ou mis à jour sur l'instance locale associée. Par exemple :

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",
 "arn:aws:s3:::aws-codedeploy-us-east-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-2/*",
 "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-2/*",

```

```

 "arn:aws:s3::aws-codedeploy-eu-west-3/*",
 "arn:aws:s3::aws-codedeploy-eu-central-1/*",
 "arn:aws:s3::aws-codedeploy-eu-central-2/*",
 "arn:aws:s3::aws-codedeploy-eu-north-1/*",
 "arn:aws:s3::aws-codedeploy-eu-south-1/*",
 "arn:aws:s3::aws-codedeploy-eu-south-2/*",
 "arn:aws:s3::aws-codedeploy-il-central-1/*",
 "arn:aws:s3::aws-codedeploy-ap-east-1/*",
 "arn:aws:s3::aws-codedeploy-ap-northeast-1/*",
 "arn:aws:s3::aws-codedeploy-ap-northeast-2/*",
 "arn:aws:s3::aws-codedeploy-ap-northeast-3/*",
 "arn:aws:s3::aws-codedeploy-ap-southeast-1/*",
 "arn:aws:s3::aws-codedeploy-ap-southeast-2/*",
 "arn:aws:s3::aws-codedeploy-ap-southeast-3/*",
 "arn:aws:s3::aws-codedeploy-ap-southeast-4/*",
 "arn:aws:s3::aws-codedeploy-ap-south-1/*",
 "arn:aws:s3::aws-codedeploy-ap-south-2/*",
 "arn:aws:s3::aws-codedeploy-me-central-1/*",
 "arn:aws:s3::aws-codedeploy-me-south-1/*",
 "arn:aws:s3::aws-codedeploy-sa-east-1/*"
]
}
]
}
```

2. Appelez la [put-user-policy](#) commande en spécifiant le nom de l'utilisateur IAM (avec l'`--user-nameoption`), le nom de la politique (avec l'`--policy-nameoption`) et le chemin d'accès au document de stratégie nouvellement créé (avec l'`--policy-documentoption`). Par exemple, en supposant que le fichier **CodeDeploy-OnPrem-Permissions.json** se trouve dans le répertoire (dossier) à partir duquel vous appelez cette commande :

### Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

## Pour attribuer des autorisations (console)

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, choisissez Politiques, puis Create Policy. (Si un bouton Mise en route est affiché, choisissez-le, puis choisissez Créer une stratégie.)
3. En regard de Create Your Own Policy (Créez votre politique), choisissez Select (Sélectionner).
4. Dans la zone Nom de la stratégie, tapez un nom pour cette stratégie (par exemple, **CodeDeploy-OnPrem-Permissions**).
5. Dans le champ Document de politique, tapez ou collez l'expression d'autorisation suivante, qui permet de AWS CodeDeploy déployer des révisions d'applications depuis n'importe quel compartiment Amazon S3 spécifié dans la politique vers l'instance sur site au nom de l'utilisateur IAM :

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

6. Choisissez Create Policy (Créer une politique).
7. Dans le panneau de navigation, choisissez utilisateurs.
8. Dans la liste des utilisateurs, recherchez et choisissez le nom de l'utilisateur IAM dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) lequel vous l'avez créé.
9. Sous l'onglet Permissions, dans Managed Policies, choisissez Attach Policy.
10. Sélectionnez la stratégie nommée **CodeDeploy-OnPrem-Permissions**, puis choisissez Attacher la stratégie.

## Étape 3 : obtenir les informations d'identification de l'utilisateur IAM

Obtenez l'ID de clé secrète et la clé d'accès secrète pour l'utilisateur IAM. Vous en aurez besoin pour l'[Étape 4 : ajouter un fichier de configuration à l'instance locale](#). Vous pouvez utiliser la console AWS CLI ou la console IAM pour obtenir l'ID de clé secrète et la clé d'accès secrète.

### Note

Si vous possédez déjà l'ID de clé secrète et la clé d'accès secrète, ignorez cette étape et accédez directement à l'[Étape 4 : ajouter un fichier de configuration à l'instance locale](#). Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS. Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

| Quel utilisateur a besoin d'un accès programmatique ? | À                                                                                                                                                                                                          | Méthode                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IAM                                                   | (Recommandé) Utilisez les informations d'identification de la console comme informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs. | <p>Suivez les instructions de l'interface que vous souhaitez utiliser.</p> <ul style="list-style-type: none"><li>• Pour le AWS CLI, voir <a href="#">Connexion pour le développement AWS local</a> dans le guide de AWS Command Line Interface l'utilisateur.</li><li>• Pour AWS SDKs, voir <a href="#">Connexion pour le développement AWS local</a> dans le guide de référence AWS SDKs and Tools.</li></ul> |



| Quel utilisateur a besoin d'un accès programmatique ?                            | À                                                                                                                                           | Méthode                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identité de la main-d'œuvre<br><br>(Utilisateurs gérés dans IAM Identity Center) | Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs. | <p>Suivez les instructions de l'interface que vous souhaitez utiliser.</p> <ul style="list-style-type: none"><li>• Pour le AWS CLI, voir <a href="#">Configuration du AWS CLI à utiliser AWS IAM Identity Center</a> dans le guide de AWS Command Line Interface l'utilisateur.</li><li>• Pour AWS SDKs, outils, et AWS APIs, voir <a href="#">Authentification IAM Identity Center</a> dans le guide de référence AWS SDKs et Tools.</li></ul> |
| IAM                                                                              | Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs. | Suivez les instructions de la section <a href="#">Utilisation d'informations d'identification temporaires avec AWS les ressources</a> du Guide de l'utilisateur IAM.                                                                                                                                                                                                                                                                            |

| Quel utilisateur a besoin d'un accès programmatique ? | À                                                                                                                                                                | Méthode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IAM                                                   | (Non recommandé)<br>Utilisez des informations d'identification à long terme pour signer des demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs. | <p>Suivez les instructions de l'interface que vous souhaitez utiliser.</p> <ul style="list-style-type: none"><li>• Pour le AWS CLI, voir <a href="#">Authentification à l'aide des informations d'identification utilisateur IAM</a> dans le Guide de l'AWS Command Line Interface utilisateur.</li><li>• Pour les outils AWS SDKs et, voir <a href="#">Authentifier à l'aide d'informations d'identification à long terme</a> dans le guide de référence des outils AWS SDKs et.</li><li>• Pour AWS APIs, voir <a href="#">Gestion des clés d'accès pour les utilisateurs IAM</a> dans le Guide de l'utilisateur IAM.</li></ul> |

Pour obtenir les informations d'identification (interface CLI)

1. Appelez la [list-access-keys](#) commande en spécifiant le nom de l'utilisateur IAM (avec l'--user-name option) et en demandant uniquement la clé d'accès IDs (avec les --output options --query et). Par exemple :

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
"AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Si aucune clé n'apparaît dans la sortie ou si des informations concernant une seule clé apparaissent dans la sortie, appelez la [create-access-key](#) commande en spécifiant le nom de l'utilisateur IAM (avec l'`--user-name` option) :

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

Dans la sortie de l'appel de la commande `create-access-key`, notez la valeur des champs `AccessKeyId` et `SecretAccessKey`. Vous aurez besoin de ces informations dans l'[Étape 4 : ajouter un fichier de configuration à l'instance locale](#).

 Important

Ce sera la seule fois que vous aurez accès à cette clé d'accès secrète. Si vous oubliez ou perdez cette clé d'accès secrète, vous devrez en générer une nouvelle en suivant la procédure dans l'[Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#).

3. Si deux clés d'accès sont déjà répertoriées, vous devez supprimer l'une d'entre elles en appelant la [delete-access-key](#) commande, en spécifiant le nom de l'utilisateur IAM (avec l'`--user-name` option) et l'ID de la clé d'accès à supprimer (avec l'`--access-key-id` option). Ensuite, appelez la commande `create-access-key`, comme indiqué plus tôt dans cette étape. Voici un exemple d'appel de la commande `delete-access-key` :

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-
key-ID
```

 Important

Si vous appelez la `delete-access-key` commande pour supprimer l'une de ces clés d'accès et qu'une instance locale utilise déjà cette clé d'accès comme décrit dans [Étape 4 : ajouter un fichier de configuration à l'instance locale](#), vous devrez suivre à [Étape 4 : ajouter un fichier de configuration à l'instance locale](#) nouveau les instructions pour spécifier un identifiant de clé d'accès et une clé d'accès secrète différents associés à cet

utilisateur IAM. Sinon, tout déploiement sur cette instance sur site risque d'être bloqué dans un état en attente perpétuel ou d'échouer complètement.

Pour obtenir les informations d'identification (console)

1.
  - a. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - b. Si la liste des utilisateurs n'est pas affichée, dans le volet de navigation, choisissez Users.
  - c. Dans la liste des utilisateurs, recherchez et choisissez le nom de l'utilisateur IAM dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) lequel vous l'avez créé.
2. Sur l'onglet Informations d'identification de sécurité, si la liste ne contient qu'une seule clé ou aucune, sélectionnez Créer une clé d'accès.

Si deux clés d'accès sont répertoriées, vous devez supprimer l'une d'entre elles. Choisissez Delete à côté d'une des clés d'accès, puis choisissez Créer une clé d'accès.

 Important

Si vous choisissez Supprimer à côté de l'une de ces clés d'accès et qu'une instance locale utilise déjà cette clé d'accès comme décrit dans [Étape 4 : ajouter un fichier de configuration à l'instance locale](#), vous devrez suivre à [Étape 4 : ajouter un fichier de configuration à l'instance locale](#) nouveau les instructions pour spécifier un identifiant de clé d'accès et une clé d'accès secrète différents associés à cet utilisateur IAM. Sinon, tout déploiement sur cette instance sur site risque d'être bloqué dans un état en attente perpétuel ou d'échouer complètement.

3. Choisissez Show et notez l'ID de clé d'accès et la clé d'accès secrète. Vous aurez besoin de ces informations à l'étape suivante. Sinon, vous pouvez choisir Download .csv file pour enregistrer une copie de l'ID de clé d'accès et de la clé d'accès secrète.

 Important

A moins que vous notiez ou téléchargiez les informations d'identification, ce sera la seule fois que vous aurez accès à cette clé d'accès secrète. Si vous oubliez ou perdez cette clé d'accès secrète, vous devrez en générer une nouvelle en suivant la procédure dans [l'Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#).

4. Choisissez Fermer pour revenir à la *IAM User Name* page Utilisateurs >.

#### Étape 4 : ajouter un fichier de configuration à l'instance locale

Ajoutez un fichier de configuration à l'instance sur site en utilisant les autorisations racine ou d'administrateur. Ce fichier de configuration sera utilisé pour déclarer les informations d'identification de l'utilisateur IAM et la AWS région cible à CodeDeploy utiliser. Le fichier doit être ajouté à un emplacement spécifique sur l'instance sur site. Le fichier doit inclure l'ARN, l'ID de clé secrète, la clé d'accès secrète et la AWS région cible de l'utilisateur IAM. Le fichier doit respecter un format spécifique.

1. Créez un fichier nommé `codedeploy.onpremises.yml` (pour un serveur Ubuntu ou une instance locale RHEL) ou `conf.onpremises.yml` (pour une instance locale Windows Server) à l'emplacement suivant sur l'instance locale :
  - Pour Ubuntu Server : `/etc/codedeploy-agent/conf`
  - Pour Windows Server : `C:\ProgramData\Amazon\CodeDeploy`
2. Utilisez un éditeur de texte pour ajouter les informations suivantes au fichier `codedeploy.onpremises.yml` ou `conf.onpremises.yml` nouvellement créé :

```

aws_access_key_id: secret-key-id
aws_secret_access_key: secret-access-key
iam_user_arn: iam-user-arn
region: supported-region
```

Où :

- *secret-key-id* est l'ID de clé secrète de l'utilisateur IAM correspondant que vous avez noté dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) ou [Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#).
- *secret-access-key* est la clé d'accès secrète de l'utilisateur IAM correspondant que vous avez notée dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) ou [Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#).
- *iam-user-arn* est l'ARN de l'utilisateur IAM que vous avez indiqué précédemment. [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#)

- ***supported-region*** est l'identifiant d'une région prise en charge par CodeDeploy laquelle se trouvent vos CodeDeploy applications, vos groupes de déploiement et vos révisions d'applications (par exemple, `us-west-2`). Pour obtenir la liste des régions, voir [Région et points de terminaison](#) dans le Références générales AWS.

#### Important

Si vous avez choisi Supprimer à côté de l'une des clés d'accès et que votre instance locale utilise déjà l'ID de clé d'accès et la clé d'accès secrète associés, vous devrez suivre les instructions [Étape 4 : ajouter un fichier de configuration à l'instance locale](#) pour spécifier un ID de clé d'accès et une clé d'accès secrète différents associés à cet utilisateur IAM. [Étape 3 : obtenir les informations d'identification de l'utilisateur IAM](#) Sinon, tout déploiement sur votre instance sur site risque d'être bloqué dans un état en attente perpétuel ou d'échouer complètement.

## Étape 5 : Installation et configuration du AWS CLI

Installez et configurez l'instance AWS CLI sur site. (Le AWS CLI sera utilisé [Étape 7 : Installation de l'CodeDeploy agent](#) pour télécharger et installer l' CodeDeploy agent sur l'instance locale.)

1. Pour installer l'instance AWS CLI sur site, suivez les instructions de la [section Configuration avec le guide AWS CLI de l'AWS Command Line Interface utilisateur](#).

#### Note

CodeDeploy les commandes permettant de travailler avec des instances locales sont devenues disponibles dans la version 1.7.19 du. AWS CLI Si vous avez une version AWS CLI déjà installée, vous pouvez vérifier sa version en appelant `aws --version`.

2. Pour configurer l'instance AWS CLI sur site, suivez les instructions de la [section Configuration de l'instance AWS CLI dans le guide de l'AWS Command Line Interface utilisateur](#).

#### Important

Lorsque vous configurez le AWS CLI (par exemple, en appelant la `aws configure` commande), veillez à spécifier l'ID de clé secrète et la clé d'accès secrète d'un utilisateur

IAM disposant au minimum des autorisations d' AWS accès suivantes en plus des autorisations d'accès spécifiées dans le [Conditions préalables à la configuration d'une instance sur site](#). Cela vous permet de télécharger et d'installer l' CodeDeploy agent sur l'instance locale :

JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*"
],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "arn:aws:s3:::amzn-s3-demo-bucket1/*"
]
 }
]
}
```

JSON

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

```
],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "*"
]
 }
]
```

Ces autorisations d'accès peuvent être attribuées soit à l'utilisateur IAM que vous avez créé, [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) soit à un autre utilisateur IAM. Pour attribuer ces autorisations à un utilisateur IAM, suivez les instructions de la section [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#), en utilisant ces autorisations d'accès au lieu de celles de cette étape.

## Étape 6 : définir la variable d' AWS\_REGION Environnement (Ubuntu Server et RHEL uniquement)

Si vous n'exécutez pas Ubuntu Server ou RHEL sur votre instance locale, ignorez cette étape et passez directement à. [Étape 7 : Installation de l' CodeDeploy agent](#)

Installez l' CodeDeploy agent sur un serveur Ubuntu ou une instance RHEL sur site et autorisez l'instance à mettre à jour l' CodeDeploy agent chaque fois qu'une nouvelle version est disponible. Pour ce faire, définissez la variable d'environnement AWS\_REGION sur l'instance en spécifiant l'identifiant d'une des régions prises en charge par CodeDeploy. Nous vous recommandons de définir la valeur en fonction de la région dans laquelle se trouvent vos CodeDeploy applications, groupes de déploiement et révisions d'applications (par exemple, us-west-2). Pour obtenir la liste des régions, voir [Région et points de terminaison](#) dans le Références générales AWS.

Pour définir la variable d'environnement, appelez la commande suivante depuis le terminal :

```
export AWS_REGION=supported-region
```



Où se *supported-region* trouve l'identifiant de région (par exemple, `us-west-2`).

## Étape 7 : Installation de l' CodeDeploy agent

Installez l' CodeDeploy agent sur l'instance locale :

- Pour une instance locale d'Ubuntu Server, suivez les instructions fournies [Installation de l' CodeDeploy agent pour Ubuntu Server](#), puis revenez à cette page.
- Pour une instance RHEL sur site, suivez les instructions fournies [Installation de l' CodeDeploy agent pour Amazon Linux ou RHEL](#), puis revenez à cette page.
- Pour une instance locale de Windows Server, suivez les instructions fournies [Installation de l' CodeDeploy agent pour Windows Server](#), puis revenez à cette page.

## Étape 8 : enregistrer l'instance locale auprès de CodeDeploy

Les instructions de cette étape supposent que vous inscrivez l'instance sur site à partir de l'instance sur site elle-même. Vous pouvez enregistrer une instance sur site à partir d'un appareil ou d'une instance distinct sur lequel elle est AWS CLI installée et configurée, comme décrit dans [Étape 5 : Installation et configuration du AWS CLI](#).

Utilisez le AWS CLI pour enregistrer l'instance sur site CodeDeploy afin qu'elle puisse être utilisée dans des déploiements.

1. Avant de pouvoir utiliser le AWS CLI, vous aurez besoin de l'ARN de l'utilisateur IAM dans [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) lequel vous l'avez créé. Si vous ne possédez pas encore l'ARN de l'utilisateur, appelez la commande [get-user](#), en spécifiant le nom de l'utilisateur IAM (avec l'option `--user-name`) et en demandant uniquement l'ARN de l'utilisateur (avec les options `--query` et `--output`) :

```
aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
```

2. Appelez la commande [register-on-premises-instance](#), en spécifiant les éléments suivants :
  - Un nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-name`).

### Important

Pour identifier plus facilement l'instance sur site, notamment à des fins de débogage, nous vous recommandons vivement de spécifier un nom faisant référence à une

caractéristique unique de l'instance sur site (par exemple, le numéro de série ou un identificateur de ressource interne unique, le cas échéant). Si vous spécifiez une adresse MAC comme nom, sachez que les adresses MAC contiennent des CodeDeploy caractères interdits, tels que deux points (:). Pour obtenir la liste des caractères autorisés, consultez la page [CodeDeploy quotas](#).

- L'ARN de l'utilisateur IAM que vous avez créé [Étape 1 : créer un utilisateur IAM pour l'instance sur site](#) (avec l'option `--iam-user-arn`).

Par exemple :

```
aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
```

## Étape 9 : baliser l'instance locale

Vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour étiqueter l'instance locale. (CodeDeploy utilise des balises d'instance locales pour identifier les cibles de déploiement lors d'un déploiement.)

Pour baliser l'instance sur site (interface CLI)

- Appelez la commande [add-tags-to-on-premises-instances](#) en spécifiant :
  - Le nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-names`).
  - Le nom de la clé de balise et de la valeur de balise de l'instance sur site que vous voulez utiliser (avec l'option `--tags`). Vous devez spécifier à la fois un nom et une valeur. CodeDeploy n'autorise pas les balises d'instance locales contenant uniquement des valeurs.

Par exemple :

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

## Pour baliser l'instance sur site (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. CodeDeploy Dans le menu, choisissez Instances locales.
3. Dans la liste des instances sur site, choisissez la flèche en regard de l'instance sur site que vous souhaitez baliser.
4. Dans la liste des balises, sélectionnez ou entrez la clé de balise et la valeur de balise souhaitées. Une fois que vous avez entré la clé de balise et la valeur de balise, une autre ligne apparaît. Vous pouvez répéter cela pour un maximum de 10 balises. Pour supprimer une balise, choisissez l'icône de suppression ( ).
5. Après avoir ajouté les balises, choisissez Update Tags.

## Étape 10 : Déployer les révisions de l'application sur l'instance locale

Vous êtes maintenant prêt à déployer les révisions d'application sur l'instance sur site inscrite et balisée.

Vous déployez des révisions d'applications sur des instances sur site de la même manière que le déploiement de révisions d'applications sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Créez un déploiement avec CodeDeploy](#). Ces instructions incluent un lien vers des prérequis, y compris la création d'une application, la création d'un groupe de déploiement et la préparation d'une révision d'application. Si vous avez besoin d'un exemple simple de révision d'application à déployer, vous pouvez créer celui décrit dans [Étape 2 : Création d'un exemple de révision d'application](#), dans la rubrique [Tutoriel : Déployer une application sur une instance locale avec CodeDeploy \(Windows Server, Ubuntu Server ou Red Hat Enterprise Linux\)](#).

### Important

Si vous réutilisez un rôle de CodeDeploy service dans le cadre de la création d'un groupe de déploiement qui cible les instances locales, vous Tag : get \* devez inclure dans la Action

partie de la déclaration de politique du rôle de service. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

## Étape 11 : suivre les déploiements vers l'instance sur site

Après avoir déployé une révision d'application sur les instances sur site inscrites et balisées, vous pouvez suivre la progression du déploiement.

Vous suivez les déploiements sur des instances sur site de la même manière que le suivi des déploiements sur des instances Amazon EC2. Pour obtenir des instructions, veuillez consulter [Afficher les détails CodeDeploy du déploiement](#).

## Gestion des opérations des instances sur site dans CodeDeploy

Suivez les instructions de cette section pour gérer les opérations sur vos instances locales une fois que vous les avez enregistrées CodeDeploy, par exemple pour obtenir plus d'informations sur les instances locales, en supprimer des balises, ainsi que pour désinstaller et annuler l'enregistrement des instances locales.

### Rubriques

- [Obtenir des informations sur une seule instance sur site](#)
- [Obtenir des informations sur plusieurs instances locales](#)
- [Supprimer manuellement les balises d'instance sur site d'une instance sur site](#)
- [Désinstallez automatiquement l' CodeDeploy agent et supprimez le fichier de configuration d'une instance locale](#)
- [Désenregistrer automatiquement une instance locale](#)
- [Désenregistrer manuellement une instance locale](#)

### Obtenir des informations sur une seule instance sur site

Vous pouvez obtenir des informations sur une instance sur site individuelle en suivant les instructions de la rubrique [Afficher les détails CodeDeploy du déploiement](#). Vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour obtenir plus d'informations sur une seule instance locale.

Pour obtenir des informations sur une instance sur site individuelle (interface CLI)

- Appelez la commande [get-on-premises-instance](#), en spécifiant le nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-name`) :

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Pour obtenir des informations sur une instance sur site individuelle (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez On-premises instances.
3. Dans la liste des instances sur site, choisissez le nom d'une instance sur site pour afficher ses détails.

## Obtenir des informations sur plusieurs instances locales

Vous pouvez obtenir des informations sur des instances sur site en suivant les instructions de la rubrique [Afficher les détails CodeDeploy du déploiement](#). Vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour obtenir plus d'informations sur les instances locales.

Pour obtenir des informations sur plusieurs instances sur site (interface CLI)

1. Pour obtenir la liste des noms des instances sur site, appelez la commande [list-on-premises-instances](#), en spécifiant :
  - Si vous voulez obtenir des informations sur toutes les instances sur site inscrites ou désinscrites (avec l'option `--registration-status` et Registered ou Deregistered, respectivement). Si vous omettez cela, les noms des instances sur site inscrites et désinscrites sont retournés.
  - Si vous voulez obtenir des informations uniquement sur les instances sur site balisées avec des balises d'instance sur site spécifiques (avec l'option `--tag-filters`). Pour

chaque balise d'instance sur site, spécifiez Key, Value et Type (qui devrait toujours être KEY\_AND\_VALUE). Séparez plusieurs balises d'instance sur site à l'aide d'espaces entre chaque triplet Key, Value et Type.

Par exemple :

```
aws deploy list-on-premises-instances --registration-status Registered
--tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
Key=Name,Value=CodeDeployDemo-OnPrem-Beta,Type=KEY_AND_VALUE
```

2. Pour des informations plus détaillées, appelez la commande [batch-get-on-premises-instances](#), avec les noms des instances locales (avec l'option `--instance-names`) :

```
aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
AssetTag09920444EX
```

Pour obtenir des informations sur plusieurs instances sur site (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez On-premises instances.

Des informations sur les instances sur site s'affichent.

## Supprimer manuellement les balises d'instance sur site d'une instance sur site

Généralement, vous supprimez une balise d'instance sur site à partir d'une instance sur site lorsque cette balise n'est plus utilisée ou si vous souhaitez supprimer l'instance sur site à partir de groupes de déploiement quelconques qui s'appuient sur cette balise. Vous pouvez utiliser la console AWS CLI ou la AWS CodeDeploy console pour supprimer les balises d'instance locales des instances locales.

Vous n'avez pas besoin de supprimer les balises d'instance sur site sur une instance sur site avant d'annuler son inscription.

La suppression manuelle des balises d'instance sur site d'une instance sur site n'annule pas l'inscription de l'instance. Il ne désinstalle pas l'agent CodeDeploy de l'instance. Elle ne supprime pas le fichier de configuration de l'instance. Il ne supprime pas l'utilisateur IAM associé à l'instance.

Pour annuler automatiquement l'inscription de l'instance sur site, consultez la page [Désenregistrer automatiquement une instance locale](#).

Pour annuler manuellement l'inscription de l'instance sur site, consultez la page [Désenregistrer manuellement une instance locale](#).

Pour désinstaller automatiquement l'agent CodeDeploy et supprimer le fichier de configuration de l'instance locale, consultez [Désinstallez automatiquement l'agent CodeDeploy et supprimez le fichier de configuration d'une instance locale](#).

Pour désinstaller manuellement uniquement l'agent CodeDeploy de l'instance locale, consultez [Gestion des opérations des agents CodeDeploy](#).

Pour supprimer manuellement l'utilisateur IAM associé, voir [Supprimer un utilisateur IAM de votre compte AWS](#).

Pour supprimer des balises d'instance sur site sur une instance sur site (interface CLI)

- Appelez [remove-tags-from-on-premises-instances](#) en spécifiant :
  - Les noms qui identifient de façon unique l'instance sur site (avec l'option `--instance-names`).
  - Les noms et les valeurs des balises que vous voulez supprimer (avec l'option `--tags`).

Par exemple :

```
aws deploy remove-tags-from-on-premises-instances --instance-names
AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Pour supprimer des balises d'instance sur site sur une instance sur site (console)

1. Connectez-vous à la console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez On-premises instances.
3. Dans la liste des instances sur site, choisissez le nom de l'instance sur site dont vous souhaitez supprimer les balises.
4. Dans la section Balises, choisissez Supprimer en regard de chaque balise à supprimer.
5. Après avoir supprimé les balises, choisissez Update tags.

## Désinstallez automatiquement l' CodeDeploy agent et supprimez le fichier de configuration d'une instance locale

Généralement, vous désinstallez l' CodeDeploy agent et supprimez le fichier de configuration d'une instance sur site une fois que vous n'avez plus l'intention de le déployer.

 Note

La désinstallation automatique de l' CodeDeploy agent et la suppression du fichier de configuration d'une instance locale n'annulent pas l'enregistrement d'une instance locale. Elle ne dissocie pas les balises d'instance sur site associées à l'instance sur site. Il ne supprime pas l'utilisateur IAM associé à l'instance locale.

Pour annuler automatiquement l'inscription de l'instance sur site, consultez la page [Désenregistrer automatiquement une instance locale](#).

Pour annuler manuellement l'inscription de l'instance sur site, consultez la page [Désenregistrer manuellement une instance locale](#).

Pour dissocier manuellement des balises d'instance sur site associées quelconques, consultez la section [Supprimer manuellement les balises d'instance sur site d'une instance sur site](#).

Pour désinstaller manuellement l' CodeDeploy agent de l'instance locale, consultez [Gestion des opérations des CodeDeploy agents](#).

Pour supprimer manuellement l'utilisateur IAM associé, voir [Supprimer un utilisateur IAM de votre AWS compte](#).



À partir de l'instance locale, utilisez la commande AWS CLI pour appeler la commande de [désinstallation](#).

Par exemple :

```
aws deploy uninstall
```

La commande `uninstall` exécute les opérations suivantes :

1. Arrête l' CodeDeploy agent en cours d'exécution sur l'instance locale.
2. Désinstalle l' CodeDeploy agent de l'instance locale.
3. Elle supprime le fichier de configuration sur l'instance sur site. (Pour Ubuntu Server et RHEL, il s'agit de `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`. Pour Windows Server, c'est `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

## Désenregistrer automatiquement une instance locale

Généralement, vous annulez l'inscription d'une instance sur site une fois que vous n'avez plus l'intention d'y effectuer un déploiement. Lorsque vous annulez l'inscription d'une instance sur site, même si cette instance sur site fait éventuellement partie des balises d'instance sur site d'un groupe de déploiement, l'instance sur site ne sera incluse dans aucun déploiement. Vous pouvez utiliser le AWS CLI pour annuler l'enregistrement des instances locales.

### Note

Vous ne pouvez pas utiliser la CodeDeploy console pour annuler l'enregistrement d'une instance locale. En outre, l'annulation de l'inscription d'une instance sur site supprime les balises d'instance sur site qui lui sont associées. Il ne désinstalle pas l' CodeDeploy agent de l'instance locale. Elle ne supprime pas le fichier de configuration de l'instance sur site de cette instance sur site.

Pour utiliser la CodeDeploy console pour effectuer certaines (mais pas toutes) des activités décrites dans cette section, consultez la section CodeDeploy console de [Désenregistrer manuellement une instance locale](#).

Pour dissocier manuellement des balises d'instance sur site associées quelconques, consultez la section [Supprimer manuellement les balises d'instance sur site d'une instance sur site](#).

Pour désinstaller automatiquement l' CodeDeploy agent et supprimer le fichier de configuration de l'instance locale, consultez [Désinstallez automatiquement l' CodeDeploy agent et supprimez le fichier de configuration d'une instance locale](#).

Pour désinstaller manuellement uniquement l' CodeDeploy agent de l'instance locale, consultez [Gestion des opérations des CodeDeploy agents](#).

Utilisez la commande AWS CLI pour appeler la commande de [désenregistrement](#), en spécifiant :

- Le nom qui identifie de manière unique l'instance locale à CodeDeploy (avec l' `--instance-name` option).
- Facultativement, s'il faut supprimer l'utilisateur IAM associé à l'instance locale. Le comportement par défaut est de supprimer l'utilisateur IAM. Si vous ne souhaitez pas supprimer l'utilisateur IAM associé à l'instance sur site, spécifiez l'option `--no-delete-iam-user` dans la commande.
- Facultativement, la AWS région dans laquelle l'instance locale a été enregistrée CodeDeploy (avec l' `--region` option). Il doit s'agir de l'une des régions prises en charge répertoriées dans [Région et des points de terminaison](#) dans le Références générales AWS (par exemple, `us-west-2`). Si cette option n'est pas spécifiée, la AWS région par défaut associée à l'utilisateur IAM appelant sera utilisée.

Exemple qui annule l'enregistrement d'une instance et supprime l'utilisateur :

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Exemple qui annule l'enregistrement d'une instance mais ne supprime pas l'utilisateur :

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region us-west-2
```

La commande `deregister` exécute les opérations suivantes :

1. Désenregistre l'instance locale auprès de. CodeDeploy
2. Si cela est spécifié, supprime l'utilisateur IAM associé à l'instance locale.

Après avoir annulé l'inscription d'une instance sur site :

- Elle n'apparaît plus immédiatement dans la console.

- Vous ne pouvez pas immédiatement créer une autre instance avec le même nom.

Si cette commande rencontre des erreurs, un message d'erreur s'affiche, qui décrit comment suivre manuellement les étapes restantes. Sinon, un message de réussite s'affiche, qui décrit comment appeler la commande `uninstall`.

## Désenregistrer manuellement une instance locale

Généralement, vous annulez l'inscription d'une instance sur site une fois que vous n'avez plus l'intention d'y effectuer un déploiement. Vous utilisez le AWS CLI pour désenregistrer manuellement les instances locales.

Le désenregistrement manuel d'une instance locale ne désinstalle pas l'agent. CodeDeploy Elle ne supprime pas le fichier de configuration de l'instance. Il ne supprime pas l'utilisateur IAM associé à l'instance. Elle ne supprime aucune balise associée à l'instance.

Pour désinstaller automatiquement l' CodeDeploy agent et supprimer le fichier de configuration de l'instance locale, consultez [Désinstallez automatiquement l' CodeDeploy agent et supprimez le fichier de configuration d'une instance locale](#).

Pour désinstaller manuellement uniquement l' CodeDeploy agent, voir [Gestion des opérations des CodeDeploy agents](#).

Pour supprimer manuellement l'utilisateur IAM associé, voir [Supprimer un utilisateur IAM de votre AWS compte](#).

Pour supprimer manuellement uniquement les balises d'instance sur site associées, consultez la page [Supprimer manuellement les balises d'instance sur site d'une instance sur site](#).

- Appelez la commande [deregister-on-premises-instance](#), en spécifiant le nom qui identifie de façon unique l'instance sur site (avec l'option `--instance-name`) :

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Après avoir annulé l'inscription d'une instance sur site :

- Elle n'apparaît plus immédiatement dans la console.
- Vous ne pouvez pas immédiatement créer une autre instance avec le même nom.

# Afficher les détails de l'instance avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails des instances utilisées dans un déploiement.

Pour plus d'informations sur l'utilisation des actions d' CodeDeploy API pour afficher les instances [GetDeploymentInstance](#), consultez [ListDeploymentInstances](#), et [ListOnPremisesInstances](#).

## Rubriques

- [Afficher les détails de l'instance \(console\)](#)
- [Afficher les détails de l'instance \(CLI\)](#)

## Afficher les détails de l'instance (console)

Pour afficher les informations de l'instance:

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Deployments.

### Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy est pris en charge dans ces régions uniquement.

3. Pour afficher les détails de déploiement, choisissez l'ID de déploiement de l'instance.
4. Vous pouvez afficher toutes les instances dans la section Activité des instances sur la page du déploiement.
5. Pour voir des informations sur des événements individuels du cycle de vie de déploiement d'une instance, sur la page des détails de déploiement, dans la colonne Events, choisissez View events.

 Note

Si le message Failed apparaît pour l'un des événements du cycle de vie, sur la page des détails de l'instance, choisissez Afficher les journaux EC2, Afficher dans ou les deux. Vous trouverez des conseils de dépannage dans la rubrique [Résolution des problèmes d'instance](#).

6. Si vous souhaitez obtenir plus d'informations sur une EC2 instance Amazon, choisissez l'ID de l'instance dans la colonne Instance ID.

## Afficher les détails de l'instance (CLI)

Pour utiliser le AWS CLI pour afficher les détails de l'instance, appelez la `get-deployment-instance` commande ou la `list-deployment-instances` commande.

Pour voir les détails relatifs à une instance unique, appelez la commande [get-deployment-instance](#), en spécifiant :

- L'ID de déploiement unique. Pour obtenir l'ID de déploiement, appelez la commande [list-deployments](#).
- L'ID d'instance unique. Pour obtenir l'ID d'instance, appelez la commande [list-deployment-instances](#).

Pour afficher la liste des IDs quatre instances utilisées dans un déploiement, appelez la [list-deployment-instances](#) commande en spécifiant :

- L'ID de déploiement unique. Pour obtenir l'ID de déploiement, appelez la commande [list-deployments](#).
- Facultativement, s'il faut inclure uniquement une instance spécifique IDs en fonction de son statut de déploiement. (Si ce n'est pas spécifié, toutes les instances correspondantes IDs seront répertoriées, quel que soit leur statut de déploiement.)

## CodeDeploy santé de l'instance

CodeDeploy surveille l'état de santé des instances d'un groupe de déploiement. Les déploiements échouent si le nombre d'instances saines chute sous le nombre minimal d'instances saines qui a été

spécifié pour le groupe de déploiement pendant un déploiement. Par exemple, si 85 % des instances doivent rester saines lors d'un déploiement, et que le groupe de déploiement contient 10 instances, le déploiement global échoue en cas d'échec du déploiement, même dans une seule instance. Cela est dû au fait que lorsqu'une instance est mise hors ligne pour que la dernière révision de l'application puisse être installée, le nombre d'instances saines disponibles tombe déjà à 90 %. Une instance ayant échoué, plus une autre instance hors ligne signifierait que seulement 80 % des instances sont saines et disponibles. Le déploiement global par CodeDeploy échoue.

Pour qu'un déploiement général réussisse, il est important de se souvenir que les éléments suivants doivent être vrais :

- CodeDeploy est capable de se déployer sur chaque instance du déploiement.
- Le déploiement doit réussir sur au moins une instance. Cela signifie que même si la valeur minimale d'hôtes sains est de 0, le déploiement doit réussir pour au moins une instance (c'est-à-dire qu'au moins une instance doit être saine) pour que le déploiement général réussisse.

## Rubriques

- [État de santé](#)
- [À propos du nombre minimum d'instances saines](#)
- [À propos du nombre minimum d'instances saines par zone de disponibilité](#)

## État de santé

CodeDeploy attribue deux valeurs d'état de santé à chaque instance : état de révision et état de santé de l'instance.

### État de santé de révision

L'état de santé de révision s'appuie sur la révision d'application actuellement installée sur l'instance. Il peut avoir les valeurs d'état suivantes :

- Actuel : la révision installée sur l'instance correspond à la révision pour le dernier déploiement réussi du groupe de déploiement.
- Ancien : la révision installée sur l'instance correspond à une version antérieure de l'application.
- Inconnu : la révision d'application n'a pas été installée correctement sur l'instance.

## État de l'instance

L'état de santé d'instance s'appuie sur la réussite ou non des déploiements sur une instance. Il peut avoir les valeurs suivantes :

- Sain : le dernier déploiement sur l'instance a réussi.
- Non sain : la tentative de déploiement d'une révision sur l'instance a échoué, ou une révision n'a pas encore été déployée sur l'instance.

CodeDeploy utilise l'état des révisions et l'état des instances pour planifier le déploiement sur les instances du groupe de déploiement dans l'ordre suivant :

1. État de santé d'instance non sain.
2. État de santé de révision inconnu.
3. État de santé de révision ancien.
4. État de santé de révision actuel.

Si le déploiement global réussit, la révision est mise à jour et les valeurs d'état de santé du groupe de déploiement sont mises à jour pour refléter le dernier déploiement.

- Toutes les instances actuelles qui ont eu un déploiement réussi restent actuelles. Sinon, elles deviennent inconnues.
- Toutes les instances anciennes ou inconnues qui ont eu un déploiement réussi deviennent actuelles. Sinon, elles restent anciennes ou inconnues.
- Toutes les instances saines qui ont eu un déploiement réussi restent saines. Sinon, elles deviennent non saines.
- Toutes les instances non saines qui ont eu un déploiement réussi deviennent saines. Sinon, elles restent non saines.

Si le déploiement global échoue ou est arrêté :

- L'état de santé de chaque instance sur laquelle a CodeDeploy tenté de déployer la révision de l'application est défini comme sain ou non, selon que la tentative de déploiement de cette instance a réussi ou échoué.
- Chaque instance sur laquelle CodeDeploy n'a pas tenté de déployer la révision de l'application conserve sa valeur d'intégrité actuelle.

- La révision du groupe de déploiement reste la même.

## À propos du nombre minimum d'instances saines

Le nombre minimum requis d'instances saines est défini dans le cadre d'une configuration de déploiement.

### Important

Lors d'un blue/green déploiement, la configuration de déploiement et la valeur minimale des hôtes sains s'appliquent aux instances de l'environnement de remplacement, et non à celles de l'environnement d'origine. Toutefois, lorsque l'enregistrement des instances dans l'environnement d'origine auprès de l'équilibreur de charge est annulé, le déploiement global est marqué comme ayant échoué même si l'enregistrement d'une seule instance d'origine ne peut pas être annulé.

CodeDeploy fournit trois configurations de déploiement par défaut qui utilisent couramment des valeurs d'hôte saines minimales :

| Nom de configuration de déploiement par défaut | Valeur minimale d'hôtes sains prédéfinie |
|------------------------------------------------|------------------------------------------|
| CodeDeployDefault.OneAtATime                   | 1                                        |
| CodeDeployDefault.HalfAtATime                  | 50%                                      |
| CodeDeployDefault.AllAtOnce                    | 0                                        |

Vous trouverez plus d'informations sur les configurations de déploiement par défaut dans [Utilisation des configurations de déploiement dans CodeDeploy](#).

Vous pouvez créer des configurations de déploiement personnalisées CodeDeploy pour définir vos propres valeurs minimales d'hôte sain. Vous pouvez définir ces valeurs en tant que nombres entiers ou pourcentages lorsque vous utilisez les opérations suivantes :

- Comme `minimum-healthy-hosts` lorsque vous utilisez la [create-deployment-config](#) commande dans le AWS CLI.



- Comme Value dans le type de [MinimumHealthyHosts](#) données de l' API CodeDeploy.
- Comme MinimumHealthyHosts lorsque vous l'utilisez [AWS::CodeDeploy::DeploymentConfig](#) dans un CloudFormation modèle.

CodeDeploy vous permet de spécifier un nombre minimum d'instances saines pour le déploiement à deux fins principales :

- Pour déterminer si le déploiement global réussit ou échoue. Le déploiement réussit si la révision d'application est correctement déployée au moins sur le nombre minimal d'instances saines.
- Pour déterminer le nombre d'instances qui doivent être saines au cours d'un déploiement pour permettre au déploiement de continuer.

Vous pouvez spécifier le nombre minimal d'instances saines pour votre groupe de déploiement comme un nombre d'instances ou comme un pourcentage du nombre total d'instances. Si vous spécifiez un pourcentage, au début du déploiement, il CodeDeploy convertit le pourcentage en un nombre équivalent d'instances, en arrondissant toutes les instances fractionnaires.

CodeDeploy suit l'état de santé des instances du groupe de déploiement pendant le processus de déploiement et utilise le nombre minimum d'instances saines spécifié pour le déploiement pour déterminer s'il convient de poursuivre le déploiement. Le principe de base est qu'un déploiement ne doit jamais faire descendre le nombre d'instances saines sous le nombre minimal que vous avez spécifié. La seule exception à cette règle est lorsqu'un groupe de déploiement a dès le début moins d'instances que le nombre minimal spécifié d'instances saines. Dans ce cas, le processus de déploiement ne réduit pas plus le nombre d'instances saines.

#### Note

CodeDeploy tentera de se déployer sur toutes les instances d'un groupe de déploiement, même celles qui sont actuellement dans un état arrêté. Dans le cadre du calcul de l'hôte sain minimum, une instance arrêtée a le même impact qu'une instance ayant échoué. Pour résoudre les échecs de déploiement en raison du trop grand nombre d'instances arrêtées, redémarrez les instances ou modifiez leurs balises de façon à les exclure du groupe de déploiement.

CodeDeploy lance le processus de déploiement en tentant de déployer la révision de l'application sur les instances défectueuses du groupe de déploiement. Pour chaque déploiement réussi, CodeDeploy

l'état de santé de l'instance devient sain et l'ajoute aux instances saines du groupe de déploiement. CodeDeploy compare ensuite le nombre actuel d'instances saines au nombre minimum d'instances saines spécifié.

- Si le nombre d'instances saines est inférieur ou égal au nombre minimum d'instances saines spécifié, CodeDeploy annule le déploiement pour garantir que le nombre d'instances saines ne diminue pas avec le nombre de déploiements supplémentaires.
- Si le nombre d'instances saines est supérieur d'au moins une au nombre minimum d'instances saines spécifié, CodeDeploy déploie la révision de l'application sur l'ensemble d'instances saines d'origine.

En cas d'échec d'un déploiement sur une instance saine, CodeDeploy l'état de santé de cette instance devient défaillant. Au fur et à mesure que le déploiement progresse, CodeDeploy met à jour le nombre actuel d'instances saines et le compare au nombre minimum d'instances saines spécifié. Si le nombre d'instances saines tombe au nombre minimum spécifié à un moment quelconque du processus de déploiement, CodeDeploy arrête le déploiement. Cette pratique empêche le prochain déploiement d'échouer et d'entraîner le nombre d'instances saines sous le nombre minimal spécifié.

#### Note

Assurez-vous que le nombre minimal d'instances saines, que vous spécifiez est inférieur au nombre total d'instances dans le groupe de déploiement. Si vous spécifiez une valeur de pourcentage, n'oubliez pas qu'il sera arrondi. Sinon, lorsque le déploiement commencera, le nombre d'instances saines sera déjà inférieur ou égal au nombre minimal spécifié d'instances saines, et CodeDeploy échouera immédiatement dans le cadre du déploiement global.

CodeDeploy utilise également le nombre minimum spécifié d'instances saines et le nombre réel d'instances saines pour déterminer si et comment déployer la révision de l'application sur plusieurs instances. Par défaut, CodeDeploy déploie la révision de l'application sur autant d'instances que possible sans risque que le nombre d'instances saines tombe en dessous du nombre minimum d'instances saines spécifié.

Pour déterminer le nombre d'instances qui doivent être déployées simultanément, CodeDeploy utilisez le calcul suivant :

$$[\text{total-hosts}] - [\text{minimum-healthy-hosts}] =$$

```
[number-of-hosts-to-deploy-to-at-once]
```

Par exemple :

- Si votre groupe de déploiement compte 10 instances et que vous définissez le nombre minimum d'instances saines à 9, CodeDeploy déploie sur une instance à la fois.
- Si votre groupe de déploiement compte 10 instances et que vous définissez le nombre minimal d'instances saines sur 3, CodeDeploy déploie simultanément sur 7 instances dans le premier lot, puis sur les 3 autres dans le second lot.
- Si votre groupe de déploiement compte 10 instances et que vous définissez le nombre minimal d'instances saines sur 0, CodeDeploy déploie simultanément sur 10 instances.

## Exemples

Dans les exemples suivants, le groupe de déploiement est supposé compter 10 instances.

Nombre minimal d'instances saines : 95 %

CodeDeploy arrondit le nombre minimum d'instances saines à 10 instances, ce qui équivaut au nombre d'instances saines. Le déploiement global échoue immédiatement sans déploiement de la révision sur aucune instance.

Nombre minimal d'instances saines : 9

CodeDeploy déploie la révision sur une instance à la fois. Si le déploiement sur l'une des instances échoue, le déploiement global échoue CodeDeploy immédiatement car le nombre d'instances saines est égal au nombre minimum d'instances saines. L'exception à cette règle est que si la dernière instance échoue, le déploiement réussit quand même.

CodeDeploy poursuit le déploiement, une instance à la fois, jusqu'à ce qu'un déploiement échoue ou que le déploiement global soit terminé. Si les 10 déploiements réussissent, le groupe de déploiement dispose désormais de 10 instances saines.

Nombre minimal d'instances saines : 8

CodeDeploy déploie la révision sur deux instances à la fois. Si deux de ces déploiements échouent, le déploiement global échoue CodeDeploy immédiatement. L'exception à cette règle est que si la dernière instance est la deuxième à échouer, le déploiement réussit quand même.

## Nombre minimal d'instances saines : 0

CodeDeploy déploie la révision dans l'ensemble du groupe de déploiement en une seule fois. Au moins un déploiement sur une instance doit réussir afin que le déploiement général réussisse. Si aucune instance n'est saine, le déploiement échoue. Cela est dû à l'exigence selon laquelle, pour qu'un déploiement global soit considéré comme réussi, au moins une instance doit être saine lorsque le déploiement global est terminé, même si la valeur minimale des instances saines est 0.

## À propos du nombre minimum d'instances saines par zone de disponibilité

### Note

Cette section utilise les termes instance et hôte de manière interchangeable pour désigner les instances Amazon EC2.

Si vous déployez sur des instances situées dans plusieurs [zones de disponibilité](#), vous pouvez éventuellement activer la [zonal configuration](#) fonctionnalité, qui permet de CodeDeploy déployer dans une zone de disponibilité à la fois.

Lorsque cette fonctionnalité est activée, elle CodeDeploy garantit que le nombre d'hôtes sains reste supérieur aux valeurs « nombre minimum d'hôtes sains par zone » et « nombre minimum d'hôtes sains ». Si le nombre d'hôtes sains tombe en dessous de l'une ou l'autre de ces valeurs, le déploiement CodeDeploy échoue dans toutes les zones de disponibilité.

Pour calculer le nombre d'hôtes sur lesquels déployer simultanément, utilisez à la fois les CodeDeploy valeurs « nombre minimal d'hôtes sains par zone » et « nombre minimal d'hôtes sains ». CodeDeploy utilisera le moindre des calculs [A] et [B], où [A] et [B] sont :

$$[A] = [\text{total-hosts}] - [\text{min-healthy-hosts}] = [\text{number-of-hosts-to-deploy-to-at-once}]$$
$$[B] = [\text{total-hosts-per-AZ}] - [\text{min-healthy-hosts-per-AZ}] = [\text{number-of-hosts-to-deploy-to-at-once-per-AZ}]$$

Après avoir déterminé le nombre d'hôtes sur lesquels déployer simultanément, effectue le CodeDeploy déploiement sur les hôtes par lots, une zone de disponibilité à la fois, avec une pause optionnelle (ou « durée de cuisson ») entre les zones.

## Exemple

Si votre déploiement est configuré comme suit :

- [total-hosts] est 200
- [minimum-healthy-hosts] est 160
- [total-hosts-per-AZ] est 100
- [minimum-healthy-hosts-per-AZ] est 50

Alors...

- $[A] = 200 - 160 = 40$
- $[B] = 100 - 50 = 50$
- 40 est inférieur à 50

Par conséquent, CodeDeploy sera déployé sur les 40 hôtes en une seule fois.

Dans ce scénario, le déploiement se déroule comme suit :

1. CodeDeploy se déploie dans la première zone de disponibilité :
  - a. CodeDeploy se déploie sur les premiers 40 hôtes.
  - b. CodeDeploy se déploie sur les 40 hôtes suivants.
  - c. CodeDeploy se déploie sur les 20 hôtes restants.

Le déploiement vers la première zone de disponibilité est maintenant terminé.

2. (Facultatif) CodeDeploy attend que le déploiement vers la première zone « démarre », tel que défini par le paramètre Durée du moniteur ou Ajouter une durée du moniteur pour la première zone. S'il n'y a aucun problème, CodeDeploy continue.
3. CodeDeploy se déploie dans la deuxième zone de disponibilité :
  - a. CodeDeploy se déploie sur les premiers 40 hôtes.
  - b. CodeDeploy se déploie sur les 40 hôtes suivants.
  - c. CodeDeploy se déploie sur les 20 hôtes restants.

Le déploiement vers la deuxième et dernière zone de disponibilité est maintenant terminé.

Pour en savoir plus sur la fonctionnalité de configuration zonale et sur la manière de spécifier le nombre minimum d'instances saines par zone de disponibilité, consultez [zonal configuration](#).

# Utilisation des configurations de déploiement dans CodeDeploy

Une configuration de déploiement est un ensemble de règles et de conditions de réussite et d'échec utilisées par CodeDeploy lors d'un déploiement. Ces règles et conditions sont différentes selon que vous déployez sur une plate-forme de calcul EC2/sur site, une plate-forme de calcul AWS Lambda ou une plate-forme de calcul Amazon ECS.

## Configurations de déploiement sur une plate-forme informatique EC2/sur site

Lorsque vous effectuez un déploiement sur une plate-forme informatique EC2/sur site, la configuration de déploiement spécifie, en utilisant une valeur « nombre minimal d'hôtes sains » et une valeur facultative « nombre minimal d'hôtes sains par zone », le nombre ou le pourcentage d'instances qui doivent rester disponibles à tout moment au cours d'un déploiement.

Vous pouvez utiliser l'une des trois configurations de déploiement prédéfinies fournies par AWS ou créer une configuration de déploiement personnalisée. Pour plus d'informations sur la création de configurations de déploiement personnalisées, consultez [Create a Deployment Configuration](#). Si vous ne spécifiez pas de configuration de déploiement, CodeDeploy utilise le CodeDeployDefault.OneAtATime configuration de déploiement.

Pour plus d'informations sur la manière de CodeDeploy surveiller et d'évaluer l'état de santé de l'instance lors d'un déploiement, consultez [Instance Health](#). Pour consulter la liste des configurations de déploiement déjà enregistrées sur votre AWS compte, consultez [View Deployment Configuration Details](#).

## Configurations de déploiement prédéfinies pour une plate-forme informatique EC2/sur site

Le tableau ci-dessous répertorie les configurations de déploiement prédéfinies.

### Note

Aucune configuration de déploiement prédéfinie ne prend en charge cette [zonal configuration](#) fonctionnalité (qui permet de spécifier le nombre d'hôtes sains par zone de disponibilité). Si

vous souhaitez utiliser cette fonctionnalité, vous devez [créer votre propre configuration de déploiement](#).

| Configuration de déploiement | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.AllAtOnce  | <p>Déploiements sur place :</p> <p>Tente de déployer une révision d'application sur autant d'instances que possible à la fois. L'état du déploiement global apparaît comme Succeeded si la révision d'application est déployée sur une ou plusieurs des instances. L'état du déploiement global apparaît comme Failed si la révision d'application n'est déployée sur aucune des instances . À l'aide d'un exemple de neuf instances , CodeDeployDefault. AllAtOnce tente de déployer simultanément sur les neuf instances . Le déploiement global réussit si le déploiement sur une seule instance réussit. Il échoue uniquement si les déploiements sur les neuf instances échouent.</p> <p>Déploiements bleu/vert :</p> <ul style="list-style-type: none"><li>• Déploiement vers un environnement de remplacement : suit les mêmes règles de déploiement que CodeDeployDefault. AllAtOnce pour les déploiements sur place.</li><li>• Réacheminement du trafic : Il réachemine le trafic vers toutes les instances dans l'environnement de remplacement à la fois. Réussit si le réacheminement du trafic vers au moins une instance réussit. Echoue après l'échec du réacheminement vers toutes les instances .</li></ul> |



| Configuration de déploiement  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.HalfAtATime | <p>Déploiements sur place :</p> <p>Effectue le déploiement au plus sur la moitié des instances à la fois (les fractions étant arrondies à l'unité inférieure). Le déploiement global réussit si la révision d'application est déployée sur au moins la moitié des instances (les fractions étant arrondies à l'unité supérieure). Sinon, le déploiement échoue. Dans l'exemple avec neuf instances , cette configuration effectue un déploiement sur quatre instances maximum à la fois. Le déploiement global réussit si le déploiement réussit sur cinq instances ou plus. Sinon, le déploiement échoue.</p> <div><p> <b>Note</b></p><p>Si vous effectuez un déploiement sur des instances appartenant à plusieurs groupes Auto Scaling, vous CodeDeploy déploierez jusqu'à la moitié des instances à la fois, quel que soit le groupe Auto Scaling auquel elles appartiennent. Supposons, par exemple, que vous disposiez de deux groupes Auto Scaling comportant chacun 10 instances. ASG1 ASG2 Dans ce scénario, CodeDeploy vous pourriez effectuer un déploiement sur 10 instances en seulement ASG1 et le considérer comme une réussite, car le déploiement a été effectué sur au moins la moitié des instances.</p></div> |

| Configuration de déploiement | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | <p>Déploiements bleu/vert :</p> <ul style="list-style-type: none"><li>• Déploiement vers un environnement de remplacement : suit les mêmes règles de déploiement que CodeDeployDefault. HalfAtATime pour les déploiements sur place.</li><li>• Réacheminement du trafic : Il réachemine le trafic vers la moitié des instances au plus dans l'environnement de remplacement à la fois. Réussit si le réacheminement vers au moins la moitié des instances réussit. Dans le cas contraire, échoue.</li></ul> |

| Configuration de déploiement | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.OneAtATime | <p>Déploiements sur place :</p> <p>Déploie la révision d'application sur une seule instance à la fois.</p> <p>Pour les groupes de déploiement qui contiennent plusieurs instances :</p> <ul style="list-style-type: none"><li>• Le déploiement global réussit si la révision d'application est déployée sur l'ensemble des instances. L'exception à cette règle est que si le déploiement sur la dernière instance échoue, le déploiement global réussit quand même. Cela est dû au CodeDeploy fait qu'une seule instance à la fois peut être mise hors ligne avec le CodeDeployDefault.OneAtATime configuration.</li><li>• Le déploiement global échoue dès que la révision d'application ne peut pas être déployée sur une instance quelconque autre que la dernière.</li><li>• Dans un exemple utilisant neuf instances , le déploiement s'effectue sur une seule instance à la fois. Le déploiement global réussit si le déploiement sur les huit premières instances réussit. Le déploiement global échoue si le déploiement sur l'une des huit premières instances échoue.</li></ul> <p>Pour les groupes de déploiement qui ne contiennent qu'une seule instance, le déploiement global réussit seulement si le déploiement sur cette instance unique réussit.</p> <p>Déploiements bleu/vert :</p> |

| Configuration de déploiement | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | <ul style="list-style-type: none"><li>• Déploiement vers un environnement de remplacement : suit les mêmes règles de déploiement que CodeDeployDefault.OneAtATime pour les déploiements sur place.</li><li>• Réacheminement du trafic : Il réachemin e le trafic vers une instance dans l'environnement de remplacement à la fois. Réussit si le réacheminement du trafic vers toutes les instances de remplacement réussit. Echoue après le premier échec de réacheminement. L'exception à cette règle est que si le déploiement sur la dernière instance échoue lors de l'inscription, le déploiement global réussit quand même.</li></ul> |

## Configurations de déploiement sur une plateforme de calcul Amazon ECS

Lorsque vous effectuez un déploiement sur une plate-forme de calcul Amazon ECS, la configuration de déploiement indique comment le trafic est transféré vers l'ensemble de tâches Amazon ECS mis à jour. Vous pouvez déplacer le trafic à l'aide d'une configuration Canary, linéaire ou de all-at-oncedéploiement. Pour de plus amples informations, veuillez consulter [Configuration de déploiement](#).

Vous pouvez aussi créer votre configuration de déploiement personnalisée linéaire ou basée sur le contrôle de validité. Pour de plus amples informations, veuillez consulter [Create a Deployment Configuration](#).

## Configurations de déploiement prédéfinies pour une plateforme de calcul Amazon ECS

Le tableau suivant répertorie les configurations prédéfinies disponibles pour les déploiements Amazon ECS.

 Note

Si vous utilisez un Network Load Balancer, seule la configuration de déploiement `CodeDeployDefault.ECSAllAtOnce` prédéfinie est prise en charge.

| Configuration de déploiement                                        | Description                                                                                               |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <code>CodeDeployDefault.ECSLinear10PercentEvery1 minutes</code>     | Déplace 10 % du trafic toutes les minutes jusqu'à ce que l'intégralité du trafic soit déplacée.           |
| <code>CodeDeployDefault.ECSLinear10PercentEvery3 minutes</code>     | Déplace 10 % du trafic toutes les trois minutes jusqu'à ce que l'intégralité du trafic soit déplacée.     |
| <code>CodeDeployDefault.ECSCanary10Pourcentage de 5 minutes</code>  | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés cinq minutes plus tard. |
| <code>CodeDeployDefault.ECSCanary10Pourcentage de 15 minutes</code> | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés 15 minutes plus tard.   |
| <code>CodeDeployDefault.ECSAllAtOnce</code>                         | Transfère immédiatement tout le trafic vers le conteneur Amazon ECS mis à jour.                           |

## Configurations de déploiement pour les déploiements CloudFormation bleu/vert (Amazon ECS)

Lorsque vous déployez sur une plate-forme de calcul Amazon ECS via des déploiements CloudFormation bleu/vert, la configuration de déploiement indique comment le trafic est transféré vers le conteneur Amazon ECS mis à jour. Vous pouvez déplacer le trafic à l'aide d'une configuration Canary, linéaire ou de all-at-oncedéploiement. Pour de plus amples informations, veuillez consulter [Configuration de déploiement](#).

Avec les déploiements CloudFormation bleu/vert, vous ne pouvez pas créer votre propre configuration de déploiement Canary ou linéaire personnalisée. Pour step-by-step obtenir des instructions sur l'utilisation CloudFormation pour gérer vos blue/green déploiements Amazon ECS, consultez la section [Automatiser les blue/green déploiements ECS par le biais de CodeDeploy l'utilisation CloudFormation](#) dans le guide de l'CloudFormation utilisateur.

 Note

La gestion blue/green des déploiements Amazon ECS avec n' CloudFormation est pas disponible dans les régions Europe (Milan), Afrique (Le Cap) et Asie-Pacifique (Osaka).

## Configurations de déploiement sur une plateforme AWS Lambda informatique

Lorsque vous effectuez un déploiement sur une plate-forme de calcul AWS Lambda, la configuration de déploiement indique la manière dont le trafic est transféré vers les nouvelles versions des fonctions Lambda de votre application. Vous pouvez déplacer le trafic à l'aide d'une configuration Canary, linéaire ou de all-at-oncedéploiement. Pour de plus amples informations, veuillez consulter [Configuration de déploiement](#).

Vous pouvez aussi créer votre configuration de déploiement personnalisée linéaire ou basée sur le contrôle de validité. Pour de plus amples informations, veuillez consulter [Create a Deployment Configuration](#).

## Configurations de déploiement prédéfinies pour une plate-forme AWS Lambda informatique

Le tableau ci-dessous répertorie les configurations de déploiement prédéfinies disponibles pour les déploiements AWS Lambda .

| Configuration de déploiement                 | Description                                                                                               |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.LambdaCanary10 % 5 minutes | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés cinq minutes plus tard. |

| Configuration de déploiement                               | Description                                                                                             |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.LambdaCanary10 pour cent 10 minutes      | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés 10 minutes plus tard. |
| CodeDeployDefault.LambdaCanary10 % 15 minutes              | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés 15 minutes plus tard. |
| CodeDeployDefault.LambdaCanary10 % 30 minutes              | Déplace 10 % du trafic dans le premier incrément. Les 90 % restants sont déployés 30 minutes plus tard. |
| CodeDeployDefault.LambdaLinear10 PercentEvery 1 minute     | Déplace 10 % du trafic toutes les minutes jusqu'à ce que l'intégralité du trafic soit déplacée.         |
| CodeDeployDefault.LambdaLinear10 à PercentEvery 2 minutes  | Déplace 10 % du trafic toutes les deux minutes jusqu'à ce que l'intégralité du trafic soit déplacée.    |
| CodeDeployDefault.LambdaLinear10 à PercentEvery 3 minutes  | Déplace 10 % du trafic toutes les trois minutes jusqu'à ce que l'intégralité du trafic soit déplacée.   |
| CodeDeployDefault.LambdaLinear10 à PercentEvery 10 minutes | Déplace 10 % du trafic toutes les 10 minutes jusqu'à ce que la totalité du trafic soit déplacée.        |
| CodeDeployDefault.LambdaAllAtOnce                          | Transfère immédiatement tout le trafic vers les fonctions Lambda mises à jour.                          |

## Rubriques

- [Create a Deployment Configuration](#)
- [View Deployment Configuration Details](#)
- [Delete a Deployment Configuration](#)

# Créez une configuration de déploiement avec CodeDeploy

Si vous ne souhaitez pas utiliser l'une des configurations de déploiement par défaut fournies CodeDeploy, vous pouvez créer la vôtre en suivant les instructions suivantes.

Vous pouvez utiliser la CodeDeploy console AWS CLI CodeDeploy APIs, le ou un CloudFormation modèle pour créer des configurations de déploiement personnalisées.

Pour plus d'informations sur l'utilisation d'un CloudFormation modèle pour créer une configuration de déploiement, consultez [CloudFormation modèles de CodeDeploy référence](#).

## Rubriques

- [Création d'une configuration de déploiement \(console\)](#)
- [Création d'une configuration de déploiement avec CodeDeploy \(AWS CLI\)](#)

## Création d'une configuration de déploiement (console)

Suivez les instructions suivantes pour créer une configuration de déploiement à l'aide de la AWS console.

Pour créer une configuration de déploiement à CodeDeploy l'aide de la console

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, sélectionnez Configurations de déploiement.

La liste des configurations de déploiement intégrées apparaît.

3. Choisissez Créer une configuration de déploiement.
4. Dans Nom de la configuration de déploiement, entrez le nom de la configuration de déploiement. Par exemple, **my-deployment-config**.
5. Sous Plateforme de calcul, sélectionnez l'une des options suivantes :



- EC2/Sur site
- AWS Lambda
- Amazon ECS

6. Effectuez l'une des actions suivantes :

- Si vous avez choisi EC2/on-premises :
  1. Sous Nombre minimal d'hôtes sains, spécifiez le nombre ou le pourcentage d'instances qui doivent rester disponibles à tout moment au cours d'un déploiement. Pour plus d'informations sur la façon dont CodeDeploy surveille et évalue l'état de l'instance lors d'un déploiement, consultez [Instance Health](#).
  2. (Facultatif) Sous Configuration zonale, sélectionnez Activer la configuration zonale pour CodeDeploy déployer votre application dans une [zone de disponibilité](#) à la fois, au sein d'une AWS région. En déployant dans une zone de disponibilité à la fois, vous pouvez exposer votre déploiement à un public de plus en plus large à mesure que la confiance dans les performances et la viabilité du déploiement augmente. Si vous n'activez pas de configuration zonale, CodeDeploy déploie votre application sur une sélection aléatoire d'hôtes dans une région.

Si vous activez la fonctionnalité de configuration zonale, notez ce qui suit :

- La fonctionnalité de configuration zonale n'est prise en charge que pour les déploiements sur place vers des instances Amazon EC2. (Les déploiements bleu/vert et les instances sur site ne sont pas pris en charge.) Pour plus d'informations sur les déploiements sur place, consultez [Type de déploiement](#).
  - La fonctionnalité de configuration zonale n'est pas prise en charge avec les [configurations de déploiement prédéfinies](#). Pour utiliser une configuration zonale, vous devez créer une configuration de déploiement personnalisée, comme décrit ici.
  - S'il est CodeDeploy nécessaire d'annuler un déploiement, CodeDeploy effectuera les opérations de restauration sur des hôtes aléatoires. (ne CodeDeploy rétrogradera pas une zone à la fois, comme on pouvait s'y attendre.) Ce comportement de restauration a été choisi pour des raisons de performances. Pour plus d'informations sur les annulations, consultez [Redéployez et annulez un déploiement avec CodeDeploy](#).
3. Si vous avez coché la case Activer la configuration zonale, spécifiez éventuellement les options suivantes :

- (Facultatif) Dans Durée du moniteur, spécifiez la période, en secondes, qui CodeDeploy doit attendre après la fin d'un déploiement dans une zone de disponibilité. CodeDeploy attendra ce laps de temps avant de commencer un déploiement vers la prochaine zone de disponibilité. Envisagez d'ajouter une durée de surveillance pour donner au déploiement le temps de faire ses preuves (ou de « cuire ») dans une zone de disponibilité avant d'être publié dans la zone suivante. Si vous ne spécifiez pas de durée de surveillance, le déploiement CodeDeploy commence immédiatement vers la zone de disponibilité suivante. Pour plus d'informations sur le fonctionnement du paramètre Durée du moniteur, consultez [À propos du nombre minimum d'instances saines par zone de disponibilité](#).
- (Facultatif) Sélectionnez Ajouter une durée de surveillance pour la première zone afin de définir une durée de surveillance qui ne s'applique qu'à la première zone de disponibilité. Vous pouvez définir cette option si vous souhaitez prolonger le temps de cuisson pour la première zone de disponibilité. Si vous ne spécifiez aucune valeur dans Ajouter une durée de surveillance de première zone, utilisez la CodeDeploy valeur de durée de surveillance pour la première zone de disponibilité.
- (Facultatif) Sous Nombre minimal d'hôtes sains par zone, spécifiez le nombre ou le pourcentage d'instances qui doivent rester disponibles par zone de disponibilité pendant un déploiement. Choisissez FLEET\_PERCENT pour spécifier un pourcentage ou HOST\_COUNT pour spécifier un nombre. Ce champ fonctionne conjointement avec le champ Nombre minimal d'hôtes sains. Pour de plus amples informations, veuillez consulter [À propos du nombre minimum d'instances saines par zone de disponibilité](#).

Si vous ne spécifiez aucune valeur sous Minimum d'hôtes sains par zone, CodeDeploy utilise une valeur par défaut de 0 pourcentage.

- Si vous avez choisi AWS LambdaAmazon ECS :
  1. Pour Type, choisissez Linear ou Canary.
  2. Dans les champs Étape et Intervalle, effectuez l'une des opérations suivantes :
    - Si vous avez choisi Canary, pour Step, entrez un pourcentage du trafic, compris entre 1 et 99, à transférer. Il s'agit du pourcentage du trafic qui est déplacé au cours du premier incrément. Le trafic restant est déplacé après l'intervalle sélectionné dans le deuxième incrément.

Pour Intervalle, entrez le nombre de minutes entre le premier et le deuxième changement de trafic.

- Si vous avez choisi Linéaire, pour Étape, entrez un pourcentage du trafic, compris entre 1 et 99, à déplacer. Il s'agit du pourcentage du trafic décalé au début de chaque intervalle.

Pour Intervalle, entrez le nombre de minutes entre chaque quart de travail incrémentiel.

## 7. Choisissez Créer une configuration de déploiement.

Vous disposez désormais d'une configuration de déploiement que vous pouvez associer à un groupe de déploiement.

## Création d'une configuration de déploiement avec CodeDeploy (AWS CLI)

Pour utiliser le AWS CLI pour créer une configuration de déploiement, appelez la [create-deployment-config](#) commande.

L'exemple suivant crée une configuration de déploiement EC2/sur site nommée `ThreeQuartersHealthy` qui nécessite que 75 % des instances cibles restent saines pendant un déploiement :

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

L'exemple suivant crée une configuration de déploiement EC2/sur site nommée `300Total150PerAZ` qui nécessite 300 instances cibles pour rester saines au total par déploiement, et 50 pour le rester par zone de disponibilité. Il définit également une durée d'affichage d'une heure.

```
aws deploy create-deployment-config --deployment-config-name 300Total150PerAZ --minimum-healthy-hosts type=HOST_COUNT,value=300 --zonal-config '{"monitorDurationInSeconds":3600,"minimumHealthyHostsPerZone":{"type":"HOST_COUNT","value":50}}'
```

L'exemple suivant crée une configuration de déploiement AWS Lambda nommée `Canary25Percent45Minutes`. Il utilise le déplacement de trafic canary pour déplacer 25 % du trafic dans le premier incrément. Les 75 % restant sont déplacés 45 minutes plus tard :

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes --traffic-routing-config
```

```
"type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

L'exemple suivant crée une configuration de déploiement Amazon ECS nommée `Canary25Percent45Minutes`. Il utilise le déplacement de trafic canary pour déplacer 25 % du trafic dans le premier incrément. Les 75 % restant sont déplacés 45 minutes plus tard :

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
--traffic-routing-config
"type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

## Consultez les détails de configuration du déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails des configurations de déploiement associées à votre AWS compte. Pour une description des configurations de CodeDeploy déploiement prédéfinies, consultez [Configurations de déploiement prédéfinies pour une plate-forme informatique EC2/sur site](#).

### Rubriques

- [Afficher les détails de configuration du déploiement \(console\)](#)
- [Afficher la configuration de déploiement \(CLI\)](#)

## Afficher les détails de configuration du déploiement (console)

Pour consulter la liste des noms de configuration de déploiement à l'aide de la CodeDeploy console, procédez comme suit :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez Deployment configurations.

Vous pouvez voir ici les noms de configuration de déploiement et les critères de chaque configuration de déploiement.

 Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy est pris en charge dans ces régions uniquement.

## Afficher la configuration de déploiement (CLI)

Pour utiliser le AWS CLI pour afficher les détails de configuration du déploiement, appelez la `get-deployment-config` commande ou la `list-deployment-configs` commande.

Pour afficher les détails d'une configuration de déploiement individuelle, appelez la commande [get-deployment-config](#), en spécifiant le nom de configuration de déploiement unique.

Pour afficher les détails relatifs à plusieurs configurations de déploiement, appelez la commande [list-deployments](#).

## Supprimer une configuration de déploiement avec CodeDeploy

Vous pouvez utiliser le AWS CLI ou CodeDeploy APIs pour supprimer les configurations de déploiement personnalisées associées à votre AWS compte. Vous ne pouvez pas supprimer les configurations de déploiement intégrées, telles que `CodeDeployDefault.AllAtOnce`, `CodeDeployDefault.HalfAtATime` et `CodeDeployDefault.OneAtATime`.

 Warning

Vous ne pouvez pas supprimer une configuration de déploiement personnalisée qui est encore en cours d'utilisation. Si vous supprimez une configuration de déploiement personnalisée inutilisée, vous ne serez plus en mesure de l'associer à de nouveaux déploiements ni à de nouveaux groupes de déploiement. Cette action ne peut pas être annulée.

AWS CLI Pour supprimer une configuration de déploiement, appelez la [delete-deployment-config](#) commande en spécifiant le nom de la configuration de déploiement. Pour afficher la liste des noms des configurations de déploiement, appelez la commande [list-deployment-configs](#).

L'exemple suivant supprime une configuration de déploiement nommée ThreeQuartersHealthy.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

# Utilisation d'applications dans CodeDeploy

Après avoir configuré des instances, mais avant de pouvoir déployer une révision, vous devez créer une application dans CodeDeploy. Une application est simplement un nom ou un conteneur utilisé par CodeDeploy pour garantir que la révision, la configuration de déploiement et le groupe de déploiement corrects sont référencés au cours d'un déploiement.

Utilisez les informations du tableau ci-dessous pour les étapes suivantes :

| Plateforme de calcul                  | Scénario                                                                                                                                            | Informations pour l'étape suivante                                                                     |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| EC2/Sur site                          | Je n'ai pas encore créé d'instances.                                                                                                                | Consultez la page <a href="#">Utilisation d'instances pour CodeDeploy</a> , puis revenez à cette page. |
| EC2/Sur site                          | J'ai créé des instances, mais je n'ai pas terminé leur balisage.                                                                                    | Consultez la page <a href="#">Tagging Instances for Deployments</a> , puis revenez à cette page.       |
| EC2/Sur site,AWS Lambda et Amazon ECS | Je n'ai pas encore créé d'application.                                                                                                              | Consultez <a href="#">Créez une application avec CodeDeploy</a> .                                      |
| EC2/Sur site,AWS Lambda et Amazon ECS | J'ai déjà créé une application, mais je n'ai pas créé de groupe de déploiement.                                                                     | Consultez <a href="#">Créez un groupe de déploiement avec CodeDeploy</a> .                             |
| EC2/Sur site,AWS Lambda et Amazon ECS | J'ai déjà créé une application et un groupe de déploiement, mais je n'ai pas créé de révision d'application.                                        | Consultez <a href="#">Travailler avec les révisions d'applications pour CodeDeploy</a> .               |
| EC2/Sur site,AWS Lambda et Amazon ECS | J'ai déjà créé une application et un groupe de déploiement, et j'ai déjà chargé ma révision d'application. Je suis prêt à effectuer le déploiement. | Consultez <a href="#">Créez un déploiement avec CodeDeploy</a> .                                       |

## Rubriques

- [Créez une application avec CodeDeploy](#)
- [Consultez les détails de l'application avec CodeDeploy](#)
- [Création d'une règle de notification](#)
- [Renommer une application CodeDeploy](#)
- [Supprimer une application dans CodeDeploy](#)

## Créez une application avec CodeDeploy

Une application est simplement un nom ou un conteneur utilisé CodeDeploy pour garantir que la révision, la configuration de déploiement et le groupe de déploiement corrects sont référencés lors d'un déploiement. Vous pouvez utiliser la CodeDeploy console, le AWS CLI CodeDeploy APIs, ou un CloudFormation modèle pour créer des applications.

Votre code, ou révision d'application, est installé sur les instances par le biais d'un processus appelé déploiement. CodeDeploy prend en charge deux types de déploiements :

- **Déploiement sur place** : l'application de chaque instance du groupe de déploiement est arrêtée, la dernière révision de l'application est installée et la nouvelle version de l'application est démarrée et validée. Vous pouvez utiliser un équilibreur de charge afin que chaque instance soit désenregistrée pendant son déploiement, puis remise en service une fois le déploiement terminé. Seuls les déploiements qui utilisent la plate-forme informatique EC2/sur site peuvent utiliser des déploiements sur place. Pour plus d'informations sur les déploiements sur place, consultez [Vue d'ensemble d'un déploiement sur place](#).
- **Déploiement bleu/vert** : le comportement de votre déploiement dépend de la plateforme informatique que vous utilisez :
  - **Blue/green on an EC2/On-Plateforme de calcul sur site** : les instances d'un groupe de déploiement (l'environnement d'origine) sont remplacées par un ensemble d'instances différent (l'environnement de remplacement) en suivant les étapes suivantes :
    - Les instances sont provisionnées pour l'environnement de remplacement.
    - La dernière version de l'application est installée sur les instances de remplacement.
    - Un temps d'attente facultatif se produit pour des activités telles que les tests d'applications et la vérification du système.
    - Les instances de l'environnement de remplacement sont enregistrées auprès d'un ou de plusieurs équilibreurs de charge Elastic Load Balancing, ce qui entraîne le reroutage du trafic



vers celles-ci. Les instances de l'environnement d'origine sont désenregistrées et peuvent être résiliées ou maintenues en cours d'exécution pour d'autres utilisations.

 **Note**

Si vous utilisez une plate-forme de calcul EC2/sur site, sachez que les blue/green déploiements fonctionnent uniquement avec les instances Amazon EC2.

- Bleu/vert sur une plateforme de calcul AWS Lambda Amazon ECS ou Amazon ECS : le trafic est décalé par incréments en fonction d'une configuration canarienne, linéaire ou all-at-once de déploiement.
- Déploiements bleu/vert via CloudFormation : le trafic est transféré de vos ressources actuelles vers vos ressources mises à jour dans le cadre d'une CloudFormation mise à jour de la pile. Actuellement, seuls les blue/green déploiements ECS sont pris en charge.

Pour plus d'informations sur les blue/green déploiements, consultez [Vue d'ensemble d'un blue/green déploiement](#).

Lorsque vous utilisez la CodeDeploy console pour créer une application, vous configurez simultanément son premier groupe de déploiement. Lorsque vous utilisez le AWS CLI pour créer une application, vous créez son premier groupe de déploiement lors d'une étape distincte.

Pour consulter la liste des applications déjà enregistrées sur votre AWS compte, consultez [Consultez les détails de l'application avec CodeDeploy](#). Pour plus d'informations sur l'utilisation d'un CloudFormation modèle pour créer une application, consultez [CloudFormation modèles de CodeDeploy référence](#).

Les deux types de déploiement ne s'appliquent pas à toutes les destinations. Le tableau suivant indique quels types de déploiement fonctionnent pour des déploiements aux trois types de destinations de déploiement.

| Destination du déploiement | Sur place | Bleu/vert |
|----------------------------|-----------|-----------|
| Amazon EC2                 | Oui       | Oui       |
| Sur site                   | Oui       | Non       |

| Destination du déploiement        | Sur place | Bleu/vert |
|-----------------------------------|-----------|-----------|
| Fonctions AWS Lambda sans serveur | Non       | Oui       |
| Applications Amazon ECS           | Non       | Oui       |

## Rubriques

- [Création d'une application pour un déploiement sur place \(console\)](#)
- [Création d'une application pour un blue/green déploiement \(console\)](#)
- [Création d'une application pour le déploiement d'un service Amazon ECS \(console\)](#)
- [Création d'une application pour le déploiement d'une AWS Lambda fonction \(console\)](#)
- [Création d'une application \(CLI\)](#)

## Création d'une application pour un déploiement sur place (console)

Pour utiliser la CodeDeploy console afin de créer une application pour un déploiement sur place :

### Warning

Ne suivez pas cette procédure si :

- Vous n'avez pas préparé vos instances pour qu'elles soient utilisées dans CodeDeploy des déploiements. Pour configurer vos instances, suivez les instructions fournies dans la rubrique [Utilisation d'instances pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous souhaitez créer une application qui utilise une configuration de déploiement personnalisée, mais vous n'avez pas encore créé la configuration de déploiement. Suivez les instructions fournies dans la rubrique [Create a Deployment Configuration](#), puis suivez la procédure décrite dans cette rubrique.
- Vous ne disposez pas d'un rôle de service qui CodeDeploy accorde la confiance et les autorisations minimales requises. Pour créer et configurer un rôle de service avec les autorisations requises, suivez les instructions de l'[Étape 2 : créer un rôle de service pour CodeDeploy](#), puis revenez aux étapes de cette rubrique.

- Vous souhaitez sélectionner un Classic Load Balancer, un Application Load Balancer ou un Network Load Balancer dans Elastic Load Balancing pour le déploiement sur place, mais vous ne l'avez pas encore créé.

Pour créer une application pour un déploiement sur place à l'aide de la CodeDeploy console :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Déployer, puis choisissez Démarrez.
3. Choisissez Créer une application.
4. Dans Nom de l'application, entrez le nom de votre application.
5. Dans Plateforme de calcul, choisissez EC2/Solutions sur site.
6. Choisissez Créer une application.
7. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement.
8. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

 Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom du groupe de déploiement, les balises, les noms des groupes Amazon EC2 Auto Scaling, ou les deux, et la configuration de déploiement), spécifiez ces paramètres sur cette page. Bien que ce nouveau groupe de déploiement et le groupe de déploiement existant portent le même nom, ils CodeDeploy sont traités comme des groupes de déploiement distincts, car ils sont chacun associés à des applications distinctes.

9. Dans Rôle de service, choisissez un rôle de service qui accorde l'accès CodeDeploy à votre instance cible.

10. Dans Type de déploiement, choisissez Sur place.
11. Dans Configuration de l'environnement, sélectionnez l'un des éléments suivants :
  - a. Groupes Amazon EC2 Auto Scaling : entrez ou choisissez le nom d'un groupe Amazon EC2 Auto Scaling dans lequel déployer la révision de votre application. Lorsque de nouvelles instances Amazon EC2 sont lancées dans le cadre d'un groupe Amazon EC2 Auto Scaling, CodeDeploy vous pouvez déployer automatiquement vos révisions sur les nouvelles instances. Vous pouvez ajouter jusqu'à 10 groupes Amazon EC2 Auto Scaling à un groupe de déploiement.
  - b. Instances Amazon EC2 ou instances sur site : dans les champs Clé et Valeur, entrez les valeurs de la paire clé-valeur que vous avez utilisée pour étiqueter les instances. Vous pouvez baliser jusqu'à 10 paires clé-valeur dans un seul groupe de balises.
    - i. Vous pouvez utiliser des caractères génériques dans le champ Valeur pour identifier toutes les instances étiquetées selon certains modèles, tels que les noms d'instances, de centres de coûts et de groupes Amazon EC2 similaires, etc. Par exemple, si vous choisissez Nom dans le champ Clé et que vous entrez **GRP-\*a** dans le champ Valeur, CodeDeploy identifie toutes les instances correspondant à ce modèle, telles que **GRP-1aGRP-2a**, et **GRP-XYZ-a**.
    - ii. Le champ Valeur est sensible à la casse.
    - iii. Pour supprimer une paire clé-valeur de la liste, choisissez Supprimer la balise.

Lorsqu' CodeDeploy il trouve des instances correspondant à chaque paire clé-valeur spécifiée ou au nom du groupe Amazon EC2 Auto Scaling, il affiche le nombre d'instances correspondantes. Choisissez le nombre pour obtenir plus d'informations sur ces instances.

Si vous souhaitez affiner les critères des instances dans lesquelles procéder au déploiement, sélectionnez Ajouter un groupe de balises pour créer un groupe de balises. Vous pouvez créer jusqu'à trois groupes de balises, comprenant chacun jusqu'à dix paires clé-valeur. Lorsque vous utilisez plusieurs groupes de balises dans un groupe de déploiement, seules les instances identifiées par tous les groupes de balises sont incluses dans le groupe de déploiement. Cela signifie qu'une instance doit correspondre à au moins une des balises dans chacun des groupes pour pouvoir être incluse dans le groupe de déploiement.

Pour plus d'informations sur l'utilisation de groupes de balises pour affiner votre groupe de déploiement, consultez la rubrique [Tagging Instances for Deployments](#).

12. Dans Paramètres de déploiement, sélectionnez une configuration de déploiement pour contrôler la vitesse à laquelle votre application est déployée vers les instances, par exemple une à la fois ou toutes à la fois. Pour plus d'informations sur les configurations de déploiement, consultez [Utilisation des configurations de déploiement dans CodeDeploy](#).
13. (Facultatif) Dans Load Balancer, sélectionnez Activer l'équilibrage de charge, puis dans les listes, sélectionnez les groupes cibles Classic Load Balancer, Application Load Balancer et Network Load Balancer pour gérer le trafic vers les instances pendant le déploiement. CodeDeploy Vous pouvez sélectionner jusqu'à 10 équilibreurs de charge classiques et 10 groupes cibles, pour un total de 20 éléments. Assurez-vous que les instances Amazon EC2 vers lesquelles vous souhaitez déployer sont enregistrées auprès des équilibreurs de charge (équilibreurs de charge classiques) ou des groupes cibles (équilibreurs de charge d'application et équilibreurs de charge réseau) sélectionnés.

Lors d'un déploiement, les instances d'origine sont désenregistrées des équilibreurs de charge et des groupes cibles sélectionnés afin d'empêcher le trafic d'être acheminé vers ces instances pendant le déploiement. Lorsque le déploiement est terminé, chaque instance est réenregistrée auprès de tous les équilibreurs de charge classiques et groupes cibles sélectionnés.

Pour plus d'informations sur les équilibreurs de charge pour les CodeDeploy déploiements, consultez [Integrating CodeDeploy with Elastic Load Balancing](#)

14. (Facultatif) Développez le mode Avancé et configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon ou les annulations automatiques.

Pour de plus amples informations, veuillez consulter [Configuration des options avancées d'un groupe de déploiement](#).

15. Choisissez Créer un groupe de déploiement.

L'étape suivante consiste à préparer une révision à déployer sur l'application et le groupe de déploiement. Pour obtenir des instructions, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

## Création d'une application pour un blue/green déploiement (console)

Pour utiliser la CodeDeploy console afin de créer une application pour un blue/green déploiement :

 Note

Un déploiement sur la plateforme de calcul AWS Lambda est toujours un blue/green déploiement. Vous ne spécifiez aucune option de type de déploiement.

 Warning

Ne suivez pas cette procédure si :

- Vous ne disposez pas d'instances sur lesquelles l' agent CodeDeploy est installé que vous souhaitez remplacer au cours du processus de blue/green déploiement. Pour configurer vos instances, suivez les instructions fournies dans la rubrique [Utilisation d'instances pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous souhaitez créer une application qui utilise une configuration de déploiement personnalisée, mais vous n'avez pas encore créé la configuration de déploiement. Suivez les instructions fournies dans la rubrique [Create a Deployment Configuration](#), puis suivez la procédure décrite dans cette rubrique.
- Vous ne disposez pas d'un rôle de service qui CodeDeploy accorde, au minimum, la confiance et les autorisations décrites dans [Étape 2 : créer un rôle de service pour CodeDeploy](#). Pour créer et configurer un rôle de service, suivez les instructions fournies dans la rubrique [Étape 2 : créer un rôle de service pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous n'avez pas créé de Classic Load Balancer, Application Load Balancer ou Network Load Balancer dans Elastic Load Balancing pour l'enregistrement des instances dans votre environnement de remplacement. Pour de plus amples informations, veuillez consulter [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#).

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Déployer, puis choisissez Démarrez.
3. Dans Nom de l'application, entrez le nom de votre application.
4. Dans Plateforme de calcul, choisissez EC2/Solutions sur site.
5. Choisissez Créer une application.
6. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement.
7. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

 Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris les balises de nom du groupe de déploiement, les noms des groupes Amazon EC2 Auto Scaling et la configuration de déploiement), choisissez ces paramètres sur cette page. Bien que ce nouveau groupe de déploiement et le groupe de déploiement existant aient le même nom, CodeDeploy les traite comme des groupes de déploiement distincts, car chacun est associé à une application distincte.

8. Dans Rôle de service, choisissez un rôle de service qui accorde l'accès CodeDeploy à votre instance cible.
9. Dans Type de déploiement, choisissez Bleu/vert.
10. Dans Configuration de l'environnement, choisissez la méthode à utiliser afin de fournir des instances pour votre environnement de remplacement :
  - a. Copier automatiquement le groupe Amazon EC2 Auto Scaling CodeDeploy : crée un groupe Amazon EC2 Auto Scaling en copiant celui que vous spécifiez.
  - b. Manually provision instances : Vous ne spécifiez pas d'instance pour l'environnement de remplacement tant que vous n'avez pas créé de déploiement. Vous devez créer les instances avant de commencer le déploiement. Au lieu de cela, vous spécifiez ici les instances que vous souhaitez remplacer.
11. En fonction de votre choix à l'étape 10, effectuez l'une des opérations suivantes:

- Si vous avez choisi Copier automatiquement le groupe Amazon EC2 Auto Scaling : dans le groupe Amazon EC2 Auto Scaling, choisissez ou entrez le nom du groupe Amazon EC2 Auto Scaling que vous souhaitez utiliser comme modèle pour le groupe Amazon EC2 Auto Scaling pour les instances de votre environnement de remplacement. Le nombre d'instances actuellement saines du groupe Amazon EC2 Auto Scaling que vous choisissez est créé dans votre environnement de remplacement.
  - Si vous avez choisi Provisionner les instances manuellement : activez les groupes Amazon EC2 Auto Scaling, les instances Amazon EC2 ou les deux pour spécifier les instances à ajouter à ce groupe de déploiement. Entrez les valeurs des balises Amazon EC2 ou les noms de groupes Amazon EC2 Auto Scaling pour identifier les instances de votre environnement d'origine (c'est-à-dire les instances que vous souhaitez remplacer ou qui exécutent la révision actuelle de l'application).
12. Dans Load balancer, sélectionnez Activer l'équilibrage de charge, puis dans les listes, sélectionnez les Classic Load Balancers, les groupes cibles Application Load Balancer et les groupes cibles Network Load Balancer auprès desquels vous souhaitez enregistrer vos instances Amazon EC2 de remplacement. Chaque instance de remplacement sera enregistrée auprès de tous les équilibreurs de charge classiques et groupes cibles sélectionnés. Vous pouvez sélectionner jusqu'à 10 équilibreurs de charge classiques et 10 groupes cibles, pour un total de 20 éléments.

Le trafic sera redirigé des instances d'origine vers les instances de remplacement conformément aux paramètres de réacheminement du trafic et de configuration de déploiement que vous avez choisis.

Pour plus d'informations sur les équilibreurs de charge pour les CodeDeploy déploiements, consultez. [Integrating CodeDeploy with Elastic Load Balancing](#)

13. Dans Paramètres de déploiement, vérifiez les options par défaut pour réacheminer le trafic vers l'environnement de remplacement, quelle configuration de déploiement utiliser pour le déploiement, et comment les instances dans l'environnement d'origine sont gérées après le déploiement.

Si vous voulez modifier les paramètres, passez à l'étape suivante. Sinon, passez à l'étape 15.

14. Pour modifier les paramètres de déploiement pour le blue/green déploiement, modifiez l'un des paramètres suivants.



| Paramètre                | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Réacheminement du trafic | <ul style="list-style-type: none"><li>• Redirigez le trafic immédiatement : dès que les instances de l'environnement de remplacement sont provisionnées et que la dernière version de l'application y est installée, elles sont automatiquement enregistrées auprès des équilibres de charge et des groupes cibles spécifiés, ce qui entraîne le reroutage du trafic vers ces instances. L'inscription des instances dans l'environnement d'origine est ensuite annulée.</li><li>• Je vais choisir de rediriger le trafic : les instances de l'environnement de remplacement ne sont pas enregistrées auprès des équilibres de charge et des groupes cibles spécifiés, sauf si vous redirigez le trafic manuellement. Si le temps d'attente que vous spécifiez s'écoule sans que le trafic soit réacheminé, le statut du déploiement devient Stopped.</li></ul> |

| Paramètre                    | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuration de déploiement | <p>Choisissez la fréquence à laquelle les instances de l'environnement de remplacement sont enregistrées auprès des équilibreurs de charge et des groupes cibles, par exemple une par une ou toutes en même temps.</p> <div> <b>Note</b><br/>Une fois le trafic acheminé avec succès vers l'environnement de remplacement, l'inscription de toutes les instances dans l'environnement d'origine est annulée en même temps quelle que soit la configuration de déploiement sélectionnée.</div> <p>Pour de plus amples informations, veuillez consulter <a href="#">Utilisation des configurations de déploiement dans CodeDeploy</a>.</p> |

| Paramètre           | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Instances d'origine | <ul style="list-style-type: none"><li>• Mettez fin aux instances d'origine du groupe de déploiement : une fois le trafic redirigé vers l'environnement de remplacement, les instances qui ont été désenregistrées des équilibres de charge et des groupes cibles sont résiliées après le délai d'attente que vous spécifiez.</li><li>• Maintenez en activité les instances d'origine du groupe de déploiement : une fois le trafic redirigé vers l'environnement de remplacement, les instances qui ont été désenregistrées des équilibres de charge et des groupes cibles continuent de fonctionner.</li></ul> |

15. (Facultatif) Dans Avancé, configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon ou les annulations automatiques.

Pour plus d'informations sur la spécification des options avancées dans les groupes de déploiement, consultez [Configuration des options avancées d'un groupe de déploiement](#).

16. Choisissez Créer un groupe de déploiement.

L'étape suivante consiste à préparer une révision à déployer sur l'application et le groupe de déploiement. Pour obtenir des instructions, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

## Création d'une application pour le déploiement d'un service Amazon ECS (console)

Vous pouvez utiliser la CodeDeploy console pour créer une application pour le déploiement d'un service Amazon ECS.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez Getting started.
3. Sur la page Créer une application, choisissez Utiliser CodeDeploy.
4. Dans Nom de l'application, entrez le nom de votre application.
5. Dans Compute Platform, choisissez Amazon ECS.
6. Choisissez Créer une application.
7. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement. Pour plus d'informations sur ce dont vous avez besoin pour créer un groupe de déploiement pour un déploiement Amazon ECS, consultez [Avant de commencer un déploiement Amazon ECS](#).
8. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

 Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom de groupe de déploiement et la configuration de déploiement), choisissez ces paramètres sur cette page. Bien que ce nouveau groupe et le groupe existant puissent porter le même nom, ils CodeDeploy sont traités comme des groupes de déploiement distincts, car chacun est associé à une application distincte.

9. Dans Rôle de service, choisissez un rôle de service qui accorde CodeDeploy l'accès à Amazon ECS. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).
10. Dans Nom de l'équilibreur de charge, choisissez le nom de l'équilibreur de charge qui achemine le trafic vers votre service Amazon ECS.
11. Dans Port de l'écouteur de production, choisissez le port et le protocole de l'écouteur qui achemine le trafic de production vers votre service Amazon ECS.
12. (Facultatif) Dans Port du récepteur de test, choisissez le port et le protocole d'un écouteur de test qui achemine le trafic vers l'ensemble de tâches de remplacement dans votre service Amazon ECS pendant le déploiement. Vous pouvez spécifier une ou plusieurs fonctions Lambda dans le AppSpec fichier qui s'exécutent pendant le hook. AfterAllowTestTraffic Les fonctions

peuvent exécuter des tests de validation. Si un test de validation échoue, une annulation du déploiement est déclenchée. Si les tests de validation réussissent, le hook suivant dans le cycle de vie du déploiement, `BeforeAllowTraffic`, est déclenché. Si aucun port d'écouteur de test n'est spécifié, rien ne se passe pendant le `AfterAllowTestTraffic` hook. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement Amazon ECS](#).

13. Dans Nom du groupe cible 1 et Nom du groupe cible 2, choisissez les groupes cibles utilisés pour acheminer le trafic lors de votre déploiement. CodeDeploy lie un groupe cible à l'ensemble de tâches d'origine de votre service Amazon ECS et l'autre à son ensemble de tâches de remplacement. Pour plus d'informations, consultez la section [Groupes cibles pour vos équilibres de charge d'application](#).

14. Choisissez Réacheminer le trafic immédiatement ou Spécifiez quand le trafic doit être redirigé pour déterminer à quel moment le rediriger vers votre service Amazon ECS mis à jour.

Si vous choisissez Réacheminer le trafic immédiatement, le déploiement redirige automatiquement le trafic une fois que l'ensemble de tâches de remplacement a été provisionné.

Si vous choisissez Spécifier quand rediriger le trafic, choisissez le nombre de jours, d'heures et de minutes à attendre une fois que l'ensemble de tâches de remplacement a été correctement provisionné. Pendant ce temps d'attente, les tests de validation dans les fonctions Lambda spécifiées dans le AppSpec fichier sont exécutés. Si le temps d'attente expire avant que le trafic ne soit redirigé, l'état du déploiement passe à `Stopped`

15. Pour la résiliation de la révision initiale, choisissez le nombre de jours, d'heures et de minutes à attendre après un déploiement réussi avant que la tâche définie initialement dans votre service Amazon ECS ne soit interrompue.
16. (Facultatif) Dans Avancé, configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon ou les annulations automatiques.

Pour de plus amples informations, veuillez consulter [Configuration des options avancées d'un groupe de déploiement](#).

# Création d'une application pour le déploiement d'une AWS Lambda fonction (console)

Vous pouvez utiliser la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

## Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez Getting started.
3. Sur la page Créer une application, choisissez Utiliser CodeDeploy.
4. Entrez le nom de votre application dans Nom de l'application.
5. Dans Plateforme de calcul, choisissez AWS Lambda.
6. Choisissez Créer une application.
7. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement.
8. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

## Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom de groupe de déploiement et la configuration de déploiement), choisissez ces paramètres sur cette page. Bien que ce nouveau groupe de déploiement et le groupe de déploiement existant puissent porter le même nom, CodeDeploy ils sont traités comme des groupes de déploiement distincts, car chacun est associé à une application distincte.

9. Dans Rôle de service, choisissez un rôle de service qui accorde CodeDeploy l'accès à AWS Lambda. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

10. Si vous souhaitez utiliser une configuration de déploiement prédéfinie, choisissez-en une dans Configuration de déploiement, puis passez à l'étape 12. Pour créer une configuration personnalisée, passez à l'étape suivante.

Pour plus d'informations sur les configurations de déploiement, consultez [Configurations de déploiement sur une plateforme AWS Lambda informatique](#).

11. Pour créer une configuration personnalisée, choisissez Créer une configuration de déploiement, puis faites ce qui suit :
  - a. Dans Nom de la configuration de déploiement, entrez un nom unique pour la configuration.
  - b. Dans Type, choisissez un type de configuration. Si vous choisissez Canary, le trafic est déplacé en deux incréments. Si vous choisissez Linear, le trafic est déplacé en incréments égaux, avec un nombre égal de minutes entre chaque incrément.
  - c. Pour Step, saisissez le pourcentage de trafic entre 1 et 99 à déplacer. Si votre type de configuration est Canary, c'est le pourcentage de trafic qui est déplacé dans le premier incrément. Le trafic restant est déplacé après l'intervalle sélectionné dans le deuxième incrément. Si votre type de configuration est Linear, c'est le pourcentage de trafic déplacé au début de chaque intervalle.
  - d. Dans Intervalle, entrez le nombre de minutes. Si votre type de configuration est Canary, ceci est le nombre de minutes entre le premier et deuxième déplacement de trafic. Pour une configuration Linéaire, il s'agit du nombre de minutes entre chaque déplacement de trafic incrémental.

 Note

La durée maximale d'un AWS Lambda déploiement est de deux jours, soit 2 880 minutes. Donc, la valeur maximale spécifiée pour Interval dans une configuration canary est de 2800 minutes. La valeur maximale d'une configuration linéaire dépend de la valeur de Step. Par exemple, si le pourcentage de trafic linéaire est de 25 %, il y aura quatre déplacements de trafic. La valeur maximale d'intervalle est 2800 divisé par quatre, soit 720 minutes.

- e. Choisissez Créer une configuration de déploiement.
12. (Facultatif) Dans Avancé, configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon ou les annulations automatiques.

Pour de plus amples informations, veuillez consulter [Configuration des options avancées d'un groupe de déploiement](#).

13. Choisissez Créer un groupe de déploiement.

## Création d'une application (CLI)

**AWS CLI** Pour créer une application, appelez la commande [create-application](#), en spécifiant un nom qui représente l'application de manière unique. (Dans un AWS compte, le nom d'une CodeDeploy application ne peut être utilisé qu'une seule fois par région. Vous pouvez réutiliser le nom d'une application dans différentes régions.)

Après avoir utilisé le AWS CLI pour créer une application, l'étape suivante consiste à créer un groupe de déploiement qui spécifie les instances vers lesquelles déployer les révisions. Pour obtenir des instructions, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#).

Une fois que vous avez créé le groupe de déploiement, l'étape suivante consiste à préparer une révision à déployer sur l'application et le groupe de déploiement. Pour obtenir des instructions, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

## Consultez les détails de l'application avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails de toutes les applications associées à votre AWS compte.

### Rubriques

- [Afficher les détails de l'application \(console\)](#)
- [Afficher les détails de l'application \(CLI\)](#)

## Afficher les détails de l'application (console)

Pour consulter les détails de l'application à l'aide de la CodeDeploy console, procédez comme suit :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.



 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy et choisissez Getting started.
3. Pour afficher des détails supplémentaires sur une application, choisissez le nom de l'application dans la liste.

## Afficher les détails de l'application (CLI)

AWS CLI Pour afficher les détails de l'application, appelez la `get-application` commande, la `batch-get-application` commande ou la `list-applications` commande.

Pour afficher les détails d'une seule application, appelez la commande [get-application](#) en spécifiant le nom de l'application.

Pour afficher des détails sur plusieurs applications, appelez la commande [batch-get-applications](#), en spécifiant plusieurs noms d'applications.

Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).

## Création d'une règle de notification

Vous pouvez utiliser des règles de notification pour informer les utilisateurs des modifications apportées aux applications de déploiement, notamment les succès et les échecs du déploiement. Les règles de notification spécifient à la fois les événements et la rubrique Amazon SNS utilisés 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 CodeDeploy.

Pour créer une règle de notification (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy/>.
2. Choisissez Application, puis une application dans laquelle vous souhaitez ajouter des notifications.

3. Sur la page de l'application, choisissez Notify (Notifier), puis Create notification rule (Créer une règle de notification). Vous pouvez également accéder à la page Settings (Paramètres) de l'application 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 CodeDeploy ou le gestionnaire de notifications.

Pour plus d'informations, voir [Comprendre le contenu et la sécurité des notifications](#).

6. Dans Événements qui déclenchent des notifications, sélectionnez les événements pour lesquels vous souhaitez envoyer des notifications.

| Catégorie   | Événements |
|-------------|------------|
| Déploiement | Échec      |
|             | Réussi(e)  |
|             | Démarré(e) |

7. Dans Targets (Cibles), choisissez Create SNS topic (Créer une rubrique SNS).

 Note

Lorsque vous créez le sujet, la politique qui permet CodeDeploy de publier des événements sur le sujet est appliquée pour vous. L'utilisation d'une rubrique créée spécifiquement pour CodeDeploy les notifications permet également de garantir que vous ajoutez uniquement des utilisateurs à la liste d'abonnement pour la rubrique pour laquelle vous souhaitez voir les notifications relatives à cette application de déploiement.

Après le préfixe codestar-notifications-, entrez un nom pour la rubrique, puis choisissez Submit (Envoyer).

 Note

Si vous souhaitez utiliser une rubrique Amazon SNS existante au lieu d'en créer une nouvelle, dans Targets (Cibles), choisissez son ARN. Assurez-vous que la rubrique dispose de la stratégie d'accès appropriée et que la liste d'abonnés contient uniquement les utilisateurs autorisés à voir des informations sur l'application de déploiement. Pour plus d'informations, consultez [Configurer les rubriques Amazon SNS existantes pour les notifications](#) et [Comprendre le contenu et la sécurité](#) des notifications.

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 ou aux canaux Slack. 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 codestar-notifications 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 inclure la ressource, les types d'événements et la cible Amazon SNS que vous souhaitez pour la règle. L'exemple suivant montre une règle de notification nommée **MyNotificationRule** d'après une application nommée *MyDeploymentApplication* 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* lorsque les déploiements sont réussis » :

```
{
 "Name": "MyNotificationRule",
 "EventTypeIds": [
 "codedeploy-application-deployment-succeeded"
],
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyDeploymentApplication",
 "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 codestar-notifications 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"
}
```

## Renommer une application CodeDeploy

Vous pouvez utiliser le AWS CLI ou CodeDeploy APIs pour modifier le nom d'une application.

Pour afficher la liste des noms d'applications, utilisez la commande AWS CLI to call the [list-applications](#).

Pour plus d'informations sur l'utilisation du AWS CLI pour modifier le nom d'une application, voir [update-application](#).

Pour plus d'informations sur l'utilisation du CodeDeploy APIs pour modifier le nom d'une application, consultez la section [API\\_UpdateApplication](#).

## Supprimer une application dans CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou une action d' CodeDeploy API pour supprimer des applications. Pour plus d'informations sur l'utilisation de CodeDeploy l'action API, consultez [DeleteApplication](#).

### Warning

La suppression d'une application supprime les informations sur l'application du système CodeDeploy, y compris toutes les informations de groupe de déploiement et tous les détails de déploiement associés. La suppression d'une application créée pour un déploiement EC2 / On-Premises ne supprime aucune révision d'application des instances ni aucune révision des compartiments Amazon S3. La suppression d'une application créée pour un déploiement EC2 / On-Premises ne met fin à aucune EC2 instance Amazon ni n'annule l'enregistrement d'une instance sur site. Cette action ne peut pas être annulée.

### Rubriques

- [Supprimer une application \(console\)](#)
- [Supprimer une application \(AWS CLI\)](#)

## Supprimer une application (console)

Pour supprimer une application à l'aide de la CodeDeploy console :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

- 
2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Dans la liste des applications, choisissez celle que vous souhaitez supprimer.

Une page contenant des informations détaillées sur l'application apparaît.

- 
- 
- 
4. Choisissez Supprimer l'application en haut à droite.
5. Lorsque vous y êtes invité, entrez **delete** pour confirmer que vous souhaitez supprimer l'application, puis choisissez Supprimer.

## Supprimer une application (AWS CLI)

AWS CLI Pour supprimer une application, appelez la commande [delete-application](#) en spécifiant le nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).

# Utilisation de groupes de déploiement dans CodeDeploy

Vous pouvez spécifier un ou plusieurs groupes de déploiement pour une CodeDeploy application. Chaque déploiement d'application utilise l'un de ses groupes de déploiement. Le groupe de déploiement contient les paramètres et les configurations utilisés lors du déploiement. La plupart des paramètres des groupes de déploiement dépendent de la plate-forme de calcul utilisée par votre application. Certains paramètres, tels que les annulations, les déclencheurs et les alarmes, peuvent être configurés pour les groupes de déploiement de n'importe quelle plate-forme informatique.

## Groupes de déploiement dans des déploiements de plateforme de calcul Amazon ECS

Dans un déploiement Amazon ECS, un groupe de déploiement spécifie le service Amazon ECS, l'équilibreur de charge, un écouteur de test facultatif et deux groupes cibles. Il indique également quand rediriger le trafic vers l'ensemble de tâches de remplacement et quand mettre fin à l'ensemble de tâches d'origine et à l'application Amazon ECS après un déploiement réussi.

## Groupes de déploiement dans les déploiements de plateformes AWS Lambda informatiques

Dans un déploiement AWS Lambda, un groupe de déploiement définit un ensemble de CodeDeploy configurations pour les futurs déploiements d'une fonction. AWS Lambda Par exemple, le groupe de déploiement indique comment acheminer le trafic vers une nouvelle version d'une fonction Lambda. Il peut également spécifier des alarmes et des restaurations. Un seul déploiement dans un groupe de déploiement AWS Lambda peut remplacer une ou plusieurs configurations de groupe.

## Groupes de déploiement dans les déploiements de plateformes de calcul EC2/sur site

Dans un déploiement EC2/sur site, un groupe de déploiement est un ensemble d'instances individuelles ciblées pour un déploiement. Un groupe de déploiement contient des instances étiquetées individuellement, des instances Amazon EC2 dans des groupes Amazon EC2 Auto Scaling, ou les deux.

Dans un déploiement sur place, les instances du groupe de déploiement sont mises à jour avec la dernière révision d'application.

Lors d'un blue/green déploiement, le trafic est redirigé d'un ensemble d'instances à un autre en annulant l'enregistrement des instances d'origine auprès d'un ou de plusieurs équilibres de charge et en enregistrant un ensemble d'instances de remplacement sur lequel la dernière version de l'application est généralement déjà installée.

Vous pouvez associer plusieurs groupes de déploiement à une application dans CodeDeploy. Cela permet de déployer une révision d'application vers différents ensembles d'instances à différents moments. Par exemple, vous pouvez utiliser un groupe de déploiement pour déployer une révision d'application dans un ensemble d'instances balisé Test dans lequel vous garantissez la qualité du code. Ensuite, vous déployez la même révision d'application vers un groupe de déploiement avec les instances balisées Staging pour la vérification supplémentaire. Enfin, lorsque vous êtes prêt à libérer la dernière application pour les clients, vous déployez dans un groupe de déploiement qui inclut les instances balisées Production.

Vous pouvez également utiliser plusieurs groupes de balises afin d'affiner encore les critères pour les instances comprises dans un groupe de déploiement. Pour plus d'informations, consultez [Tagging Instances for Deployments](#).

Lorsque vous utilisez la CodeDeploy console pour créer une application, vous configurez simultanément son premier groupe de déploiement. Lorsque vous utilisez le AWS CLI pour créer une application, vous créez son premier groupe de déploiement lors d'une étape distincte.

Pour consulter la liste des groupes de déploiement déjà associés à votre AWS compte, consultez [Afficher les détails du groupe de déploiement avec CodeDeploy](#).

Pour plus d'informations sur les balises d'instance Amazon EC2, consultez [Utilisation des balises à l'aide de la console](#). Pour en savoir plus sur les instances sur site, consultez la rubrique [Working with On-Premises Instances](#). Pour plus d'informations sur Amazon EC2 Auto Scaling, [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#) consultez.

## Rubriques

- [the section called “Création d'un groupe de déploiement”](#)
- [the section called “Afficher les détails du groupe de déploiement”](#)
- [the section called “Modifier les paramètres du groupe de déploiement”](#)
- [the section called “Configuration des options avancées d'un groupe de déploiement”](#)



- [the section called “Supprimer un groupe de déploiement”](#)

## Créez un groupe de déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console, le AWS CLI CodeDeploy APIs, ou un CloudFormation modèle pour créer des groupes de déploiement. Pour plus d'informations sur l'utilisation d'un CloudFormation modèle pour créer un groupe de déploiement, consultez [CloudFormation modèles de CodeDeploy référence](#).

Lorsque vous utilisez la CodeDeploy console pour créer une application, vous configurez son premier groupe de déploiement en même temps. Lorsque vous utilisez le AWS CLI pour créer une application, vous créez son premier groupe de déploiement lors d'une étape distincte.

Dans le cadre de la création d'un groupe de déploiement, vous devez spécifier un rôle de service. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

### Rubriques

- [Création d'un groupe de déploiement pour un déploiement sur place \(console\)](#)
- [Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site \(console\)](#)
- [Création d'un groupe de déploiement pour un déploiement Amazon ECS \(console\)](#)
- [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#)
- [Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS](#)
- [Création d'un groupe de déploiement \(CLI\)](#)

## Création d'un groupe de déploiement pour un déploiement sur place (console)

Pour utiliser la CodeDeploy console afin de créer un groupe de déploiement pour un déploiement sur place :

### Warning

Ne suivez pas cette procédure si :

- Vous n'avez pas préparé vos instances pour qu'elles soient utilisées dans le premier déploiement CodeDeploy d'une application. Pour configurer vos instances, suivez les instructions fournies dans la rubrique [Utilisation d'instances pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous souhaitez créer un groupe de déploiement qui utilise une configuration de déploiement personnalisée, mais vous n'avez pas encore créé la configuration de déploiement. Suivez les instructions fournies dans la rubrique [Create a Deployment Configuration](#), puis suivez la procédure décrite dans cette rubrique.
- Vous ne disposez pas d'un rôle de service que CodeDeploy accorde, au minimum, la confiance et les autorisations décrites dans [Étape 2 : créer un rôle de service pour CodeDeploy](#). Pour créer et configurer un rôle de service, suivez les instructions fournies dans la rubrique [Étape 2 : créer un rôle de service pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous souhaitez sélectionner un Classic Load Balancer, un Application Load Balancer ou un Network Load Balancer dans Elastic Load Balancing pour le déploiement sur place, mais vous ne l'avez pas encore créé.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Sur la page Applications, choisissez le nom de l'application pour laquelle vous voulez créer un groupe de déploiement.
4. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement.
5. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

 Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom du groupe de déploiement, les balises, les noms des groupes Amazon EC2 Auto Scaling, ou les deux, et la configuration de déploiement), spécifiez ces paramètres sur cette page. Bien que ce nouveau groupe de déploiement et le groupe de déploiement existant portent le même nom, ils CodeDeploy sont traités comme des groupes de déploiement distincts, car ils sont chacun associés à des applications distinctes.

6. Dans Rôle de service, choisissez un rôle de service qui accorde l'accès CodeDeploy à votre instance cible.
7. Dans Type de déploiement, choisissez Sur place.
8. Dans Configuration de l'environnement, procédez comme suit :
  - a. Si vous souhaitez déployer votre application dans un groupe Amazon EC2 Auto Scaling, sélectionnez les groupes Amazon EC2 Auto Scaling, puis choisissez le nom du groupe Amazon EC2 Auto Scaling dans lequel déployer la révision de votre application. Lorsque de nouvelles instances Amazon EC2 sont lancées dans le cadre d'un groupe Amazon EC2 Auto Scaling, CodeDeploy vous pouvez déployer automatiquement vos révisions sur les nouvelles instances. Vous pouvez ajouter jusqu'à 10 groupes Amazon EC2 Auto Scaling à un groupe de déploiement. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).
  - b. Si vous avez sélectionné les groupes Amazon EC2 Auto Scaling, sélectionnez éventuellement Ajouter un crochet de terminaison aux groupes Auto Scaling pour CodeDeploy installer un crochet de terminaison dans votre groupe Auto Scaling lorsque vous créez ou mettez à jour le groupe de déploiement. Lorsque ce hook est installé, il CodeDeploy effectuera des déploiements de terminaison. Pour de plus amples informations, veuillez consulter [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#).
  - c. Si vous souhaitez baliser vos instances, sélectionnez les instances Amazon EC2 ou les instances sur site. Dans les champs Clé et Valeur, entrez les valeurs de la paire clé-valeur que vous avez utilisée pour étiqueter les instances. Vous pouvez baliser jusqu'à 10 paires clé-valeur dans un seul groupe de balises.

- i. Vous pouvez utiliser des caractères génériques dans le champ Valeur pour identifier toutes les instances étiquetées selon certains modèles, tels que les noms d'instances, de centres de coûts et de groupes Amazon EC2 similaires, etc. Par exemple, si vous choisissez Nom dans le champ Clé et que vous entrez **GRP-\*a** dans le champ Valeur, CodeDeploy identifie toutes les instances correspondant à ce modèle, telles que **GRP-1aGRP-2a**, et **GRP-XYZ-a**.
- ii. Le champ Valeur est sensible à la casse.
- iii. Pour supprimer une paire clé-valeur de la liste, choisissez l'icône de suppression.

Lorsqu' CodeDeploy il trouve des instances correspondant à chaque paire clé-valeur spécifiée ou au nom du groupe Amazon EC2 Auto Scaling, il affiche le nombre d'instances correspondantes. Pour plus d'informations sur ces instances, cliquez sur le nombre.

Si vous souhaitez affiner les critères des instances dans lesquelles procéder au déploiement, sélectionnez Ajouter un groupe de balises pour créer un groupe de balises. Vous pouvez créer jusqu'à trois groupes de balises, comprenant chacun jusqu'à 10 paires clé-valeur. Lorsque vous utilisez plusieurs groupes de balises dans un groupe de déploiement, seules les instances identifiées par tous les groupes de balises sont incluses dans le groupe de déploiement. Cela signifie qu'une instance doit correspondre à au moins une des balises dans chacun des groupes pour pouvoir être incluse dans le groupe de déploiement.

Pour plus d'informations sur l'utilisation de groupes de balises pour affiner votre groupe de déploiement, consultez la rubrique [Tagging Instances for Deployments](#).

9. Dans Configuration de l'agent avec Systems Manager, spécifiez comment vous souhaitez installer et mettre à jour l' CodeDeploy agent sur les instances de votre groupe de déploiement. Pour plus d'informations sur l' CodeDeploy agent, consultez la section [Utilisation de l' CodeDeploy agent](#). Pour plus d'informations sur Systems Manager, voir [Qu'est-ce que Systems Manager ?](#)
  - a. Jamais : ignorez la configuration de CodeDeploy l'installation avec Systems Manager. L'agent doit être installé sur les instances pour être utilisé dans les déploiements. Choisissez cette option uniquement si vous souhaitez installer l'agent CodeDeploy d'une autre manière.
  - b. Une seule fois : Systems Manager installera l' CodeDeploy agent une fois sur chaque instance de votre groupe de déploiement.

- c. Maintenant et planifiez les mises à jour : Systems Manager créera une association avec State Manager qui installera l' CodeDeploy agent selon le calendrier que vous avez configuré. Pour de plus amples informations sur State Manager et les associations, veuillez consulter [À propos de State Manager](#).
10. Dans Configuration de déploiement, sélectionnez une configuration de déploiement pour contrôler la vitesse à laquelle les instances sont déployées, par exemple une à la fois ou toutes à la fois. Pour plus d'informations sur les configurations de déploiement, consultez [Utilisation des configurations de déploiement dans CodeDeploy](#).
11. (Facultatif) Dans Load Balancer, sélectionnez Activer l'équilibrage de charge, puis dans les listes, sélectionnez les groupes cibles Classic Load Balancer, Application Load Balancer et Network Load Balancer pour gérer le trafic vers les instances pendant le déploiement. CodeDeploy Vous pouvez sélectionner jusqu'à 10 équilibreurs de charge classiques et 10 groupes cibles, pour un total de 20 éléments. Assurez-vous que les instances Amazon EC2 vers lesquelles vous souhaitez déployer sont enregistrées auprès des équilibreurs de charge (équilibreurs de charge classiques) ou des groupes cibles (équilibreurs de charge d'application et équilibreurs de charge réseau) sélectionnés.

Lors d'un déploiement, les instances d'origine sont désenregistrées des équilibreurs de charge et des groupes cibles sélectionnés afin d'empêcher le trafic d'être acheminé vers ces instances pendant le déploiement. Lorsque le déploiement est terminé, chaque instance est réenregistrée auprès de tous les équilibreurs de charge classiques et groupes cibles sélectionnés.

Pour plus d'informations sur les équilibreurs de charge pour les CodeDeploy déploiements, consultez. [Integrating CodeDeploy with Elastic Load Balancing](#)

 Warning

Si vous configurez à la fois des groupes Auto Scaling et des équilibreurs de charge Elastic Load Balancing dans ce groupe de déploiement et que vous souhaitez [associer des équilibreurs de charge aux groupes Auto Scaling](#), nous vous recommandons de compléter cette pièce jointe avant de créer le CodeDeploy déploiement à partir de ce groupe de déploiement. Toute tentative de terminer la pièce jointe après avoir créé le déploiement peut entraîner le désenregistrement inattendu de toutes les instances des équilibreurs de charge.

12. (Facultatif) Développez Advanced et configurez toutes les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon, les options Auto Scaling ou les annulations automatiques.

Pour de plus amples informations, veuillez consulter [Configuration des options avancées d'un groupe de déploiement](#).

13. Choisissez Créer un groupe de déploiement.

## Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site (console)

Pour utiliser la CodeDeploy console afin de créer un groupe de déploiement pour un déploiement bleu/vert :

### Warning

Ne suivez pas cette procédure si :

- Vous ne disposez pas d'instances sur lesquelles l' CodeDeploy agent est installé que vous souhaitez remplacer au cours du processus de blue/green déploiement. Pour configurer vos instances, suivez les instructions fournies dans la rubrique [Utilisation d'instances pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous souhaitez créer une application qui utilise une configuration de déploiement personnalisée, mais vous n'avez pas encore créé la configuration de déploiement. Suivez les instructions fournies dans la rubrique [Create a Deployment Configuration](#), puis suivez la procédure décrite dans cette rubrique.
- Vous ne disposez pas d'un rôle de service que CodeDeploy accorde, au minimum, la confiance et les autorisations décrites dans [Étape 2 : créer un rôle de service pour CodeDeploy](#). Pour créer et configurer un rôle de service, suivez les instructions fournies dans la rubrique [Étape 2 : créer un rôle de service pour CodeDeploy](#), puis suivez la procédure décrite dans cette rubrique.
- Vous n'avez pas créé de Classic Load Balancer ou d'Application Load Balancer dans Elastic Load Balancing pour l'enregistrement des instances dans votre environnement de remplacement. Pour de plus amples informations, veuillez consulter [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#).

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Sur la page Applications, choisissez le nom de l'application pour laquelle vous voulez créer un groupe de déploiement.
4. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement.
5. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

 Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom du groupe de déploiement, les balises, les noms des groupes Amazon EC2 Auto Scaling et la configuration de déploiement), choisissez ces paramètres sur cette page. Bien que ce nouveau groupe de déploiement et le groupe de déploiement existant aient le même nom, CodeDeploy les traite comme des groupes de déploiement distincts, car ils sont associés à des applications distinctes.

6. Dans Rôle de service, choisissez un rôle de service qui accorde l'accès CodeDeploy à votre instance cible.
7. Dans Type de déploiement, choisissez Bleu/vert.
8. Dans Configuration de l'environnement, procédez comme suit :
  - Sélectionnez la méthode à utiliser pour fournir des instances pour votre environnement de remplacement. Vous avez les options suivantes :
    - Copier automatiquement le groupe Amazon EC2 Auto Scaling CodeDeploy : crée un groupe Amazon EC2 Auto Scaling en copiant celui que vous spécifiez.
    - Manually provision instances : Vous ne spécifiez pas d'instance pour l'environnement de remplacement tant que vous n'avez pas créé de déploiement. Vous devez créer les

instances avant de commencer le déploiement. Au lieu de cela, vous spécifiez ici les instances que vous souhaitez remplacer.

- Si vous avez sélectionné Copier automatiquement le groupe Amazon EC2 Auto Scaling, sélectionnez éventuellement Ajouter un crochet de terminaison aux groupes Auto Scaling pour CodeDeploy installer un crochet de terminaison dans votre groupe Auto Scaling lorsque vous créez ou mettez à jour le groupe de déploiement. Lorsque ce hook est installé, il CodeDeploy effectuera des déploiements de terminaison. Pour de plus amples informations, veuillez consulter [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#).
9. Dans Configuration de l'agent avec Systems Manager, spécifiez comment vous souhaitez installer et mettre à jour l' CodeDeploy agent sur les instances de votre groupe de déploiement. Pour plus d'informations sur l' CodeDeploy agent, consultez la section [Utilisation de l' CodeDeploy agent](#). Pour plus d'informations sur Systems Manager, voir [Qu'est-ce que Systems Manager ?](#)
- a. Jamais : ignorez la configuration de CodeDeploy l'installation avec Systems Manager. L'agent doit être installé sur les instances pour être utilisé dans les déploiements. Choisissez cette option uniquement si vous souhaitez installer l'agent CodeDeploy d'une autre manière.
  - b. Une seule fois : Systems Manager installera l' CodeDeploy agent une fois sur chaque instance de votre groupe de déploiement.
  - c. Maintenant et planifiez les mises à jour : Systems Manager créera une association avec State Manager qui installera l' CodeDeploy agent selon le calendrier que vous avez configuré. Pour de plus amples informations sur State Manager et les associations, veuillez consulter [À propos de State Manager](#).
10. En fonction de votre choix à l'étape 8, effectuez l'une des opérations suivantes:
- Si vous avez choisi Copier automatiquement le groupe Amazon EC2 Auto Scaling : dans le groupe Amazon EC2 Auto Scaling, choisissez ou entrez le nom du groupe Amazon EC2 Auto Scaling que vous souhaitez utiliser comme modèle pour le groupe Amazon EC2 Auto Scaling créé pour les instances de votre environnement de remplacement. Le nombre d'instances actuellement saines dans le groupe Amazon EC2 Auto Scaling que vous sélectionnez est créé dans votre environnement de remplacement.
  - Si vous avez choisi Provisionner les instances manuellement : sélectionnez les groupes Amazon EC2 Auto Scaling, les instances Amazon EC2 Auto Scaling, ou les deux pour spécifier les instances à ajouter à ce groupe de déploiement. Entrez les valeurs des balises Amazon EC2 Auto Scaling ou les noms de groupes Amazon EC2 Auto Scaling pour identifier



les instances de votre environnement d'origine (c'est-à-dire les instances que vous souhaitez remplacer ou qui exécutent la révision actuelle de l'application).

11. Dans Load balancer, sélectionnez Activer l'équilibrage de charge, puis dans les listes, sélectionnez les Classic Load Balancers, les groupes cibles Application Load Balancer et les groupes cibles Network Load Balancer auprès desquels vous souhaitez enregistrer vos instances Amazon EC2 de remplacement. Chaque instance de remplacement sera enregistrée auprès de tous les équilibreurs de charge classiques et groupes cibles sélectionnés. Vous pouvez sélectionner jusqu'à 10 équilibreurs de charge classiques et 10 groupes cibles, pour un total de 20 éléments.

Le trafic sera redirigé des instances d'origine vers les instances de remplacement conformément aux paramètres de réacheminement du trafic et de configuration de déploiement que vous avez choisis.

Pour plus d'informations sur les équilibreurs de charge pour les CodeDeploy déploiements, consultez [Integrating CodeDeploy with Elastic Load Balancing](#)

 Warning

Si vous configurez à la fois des groupes Auto Scaling et des équilibreurs de charge Elastic Load Balancing dans ce groupe de déploiement, et que vous souhaitez [associer les équilibreurs de charge à des groupes Auto Scaling](#), nous vous recommandons de compléter cette pièce jointe avant de créer le CodeDeploy déploiement à partir de ce groupe de déploiement. Toute tentative de terminer la pièce jointe après avoir créé le déploiement peut entraîner le désenregistrement inattendu de toutes les instances des équilibreurs de charge.

12. Dans Paramètres de déploiement, vérifiez les options par défaut pour réacheminer le trafic vers l'environnement de remplacement, quelle configuration de déploiement utiliser pour le déploiement, et comment les instances dans l'environnement d'origine sont gérées après le déploiement.

Si vous voulez modifier les paramètres, passez à l'étape suivante. Sinon, passez à l'étape 14.

13. Pour modifier les paramètres de déploiement pour le blue/green déploiement, choisissez l'un des paramètres suivants.

| Paramètre                | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Réacheminement du trafic | <ul style="list-style-type: none"><li>• Redirigez le trafic immédiatement : dès que les instances de l'environnement de remplacement sont provisionnées et que la dernière version de l'application y est installée, elles sont automatiquement enregistrées auprès des équilibres de charge et des groupes cibles spécifiés, ce qui entraîne le reroutage du trafic vers ces instances. L'inscription des instances dans l'environnement d'origine est ensuite annulée.</li><li>• Je vais choisir de rediriger le trafic : les instances de l'environnement de remplacement ne sont pas enregistrées auprès des équilibres de charge et des groupes cibles spécifiés, sauf si vous redirigez le trafic manuellement. Si le temps d'attente que vous spécifiez s'écoule sans que le trafic soit réacheminé, le statut du déploiement devient Stopped.</li></ul> |

| Paramètre                    | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuration de déploiement | <p>Choisissez la fréquence à laquelle les instances de l'environnement de remplacement sont enregistrées auprès des équilibreurs de charge et des groupes cibles, par exemple une par une ou toutes en même temps.</p> <div> <b>Note</b><br/>Une fois le trafic acheminé avec succès vers l'environnement de remplacement, l'inscription de toutes les instances dans l'environnement d'origine est annulée en même temps quelle que soit la configuration de déploiement sélectionnée.</div> <p>Pour de plus amples informations, veuillez consulter <a href="#">Utilisation des configurations de déploiement dans CodeDeploy</a>.</p> |

| Paramètre           | Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Instances d'origine | <ul style="list-style-type: none"> <li>• Mettez fin aux instances d'origine du groupe de déploiement : une fois le trafic redirigé vers l'environnement de remplacement, les instances qui ont été désenregistrées des équilibres de charge et des groupes cibles sont résiliées après le délai d'attente que vous spécifiez. Notez ce qui suit : <ul style="list-style-type: none"> <li>• Si la protection contre la suppression est activée (<code>prevent-force-deletion</code> ou <code>prevent-all-deletion</code>) dans votre groupe Auto Scaling, le déploiement CodeDeploy bleu/vert désactivera cette protection sur le groupe Auto Scaling d'origine (bleu) avant de le supprimer lors du nettoyage du déploiement. Le groupe Auto Scaling de remplacement (vert) conservera les mêmes paramètres de protection contre la suppression que le groupe d'origine.</li> </ul> </li> <li>• Maintenir les instances d'origine du groupe de déploiement en cours d'exécution : une fois le trafic redirigé vers l'environnement de remplacement, les instances qui ont été désenregistrées des équilibres de charge et des groupes cibles continuent de fonctionner.</li> </ul> |

14. (Facultatif) Dans **Advanced**, configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon, les options Auto Scaling ou les annulations automatiques.

Pour plus d'informations sur la spécification des options avancées dans les groupes de déploiement, consultez [Configuration des options avancées d'un groupe de déploiement](#).

15. Choisissez **Créer un groupe de déploiement**.

## Création d'un groupe de déploiement pour un déploiement Amazon ECS (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Dans la table des applications, choisissez le nom de l'application qui est associée au groupe de déploiement que vous voulez modifier.
4. Sur la page de l'application, dans la section Groupes de déploiement, choisissez le nom du groupe de déploiement que vous souhaitez modifier.
5. Sur la page de votre application, dans l'onglet Groupe de déploiement, choisissez Créer un groupe de déploiement. Pour plus d'informations sur ce dont vous avez besoin pour créer un groupe de déploiement pour un déploiement Amazon ECS, consultez [Avant de commencer un déploiement Amazon ECS](#).
6. Dans Nom du groupe de déploiement, entrez un nom décrivant le groupe de déploiement.

### Note

Si vous souhaitez utiliser les mêmes paramètres que ceux utilisés dans un autre groupe de déploiement (y compris le nom de groupe de déploiement et la configuration de déploiement), choisissez ces paramètres sur cette page. Bien que ce nouveau groupe et le groupe existant puissent porter le même nom, ils CodeDeploy sont traités comme des groupes de déploiement distincts, car chacun est associé à une application distincte.

7. Dans Rôle de service, choisissez un rôle de service qui accorde CodeDeploy l'accès à Amazon ECS. Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).
8. Dans Nom de l'équilibreur de charge, choisissez le nom de l'équilibreur de charge qui achemine le trafic vers votre service Amazon ECS.

9. Dans Port de l'écouteur de production, choisissez le port et le protocole de l'écouteur qui achemine le trafic de production vers votre service Amazon ECS.
10. (Facultatif) Dans Port du récepteur de test, choisissez le port et le protocole d'un écouteur de test qui achemine le trafic vers l'ensemble de tâches de remplacement dans votre service Amazon ECS pendant le déploiement. Vous pouvez spécifier une ou plusieurs fonctions Lambda dans le AppSpec fichier qui s'exécutent pendant le hook. `AfterAllowTestTraffic` Les fonctions peuvent exécuter des tests de validation. Si un test de validation échoue, une annulation du déploiement est déclenchée. Si les tests de validation réussissent, le hook suivant dans le cycle de vie du déploiement, `BeforeAllowTraffic`, est déclenché. Si aucun port d'écouteur de test n'est spécifié, rien ne se passe pendant le `AfterAllowTestTraffic` hook. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement Amazon ECS](#).
11. Dans Nom du groupe cible 1 et Nom du groupe cible 2, choisissez les groupes cibles utilisés pour acheminer le trafic lors de votre déploiement. CodeDeploy lie un groupe cible à l'ensemble de tâches d'origine de votre service Amazon ECS et l'autre à son ensemble de tâches de remplacement. Pour plus d'informations, consultez la section [Groupes cibles pour vos équilibres de charge d'application](#).
12. Choisissez Réacheminer le trafic immédiatement ou Spécifiez quand le trafic doit être redirigé pour déterminer à quel moment le rediriger vers votre service Amazon ECS mis à jour.

Si vous choisissez Réacheminer le trafic immédiatement, le déploiement redirige automatiquement le trafic une fois que l'ensemble de tâches de remplacement a été provisionné.

Si vous choisissez Spécifier quand rediriger le trafic, choisissez le nombre de jours, d'heures et de minutes à attendre une fois que l'ensemble de tâches de remplacement a été correctement provisionné. Pendant ce temps d'attente, les tests de validation dans les fonctions Lambda spécifiées dans le AppSpec fichier sont exécutés. Si le temps d'attente expire avant que le trafic ne soit redirigé, l'état du déploiement passe à `Stopped`.

13. Pour la résiliation de la révision initiale, choisissez le nombre de jours, d'heures et de minutes à attendre après un déploiement réussi avant que la tâche définie initialement dans votre service Amazon ECS ne soit interrompue.
14. (Facultatif) Dans Avancé, configurez les options que vous souhaitez inclure dans le déploiement, telles que les déclencheurs de notification Amazon SNS, les CloudWatch alarmes Amazon ou les annulations automatiques.

Pour de plus amples informations, veuillez consulter [Configuration des options avancées d'un groupe de déploiement](#).

## Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2

Avant d'exécuter un blue/green déploiement, ou un déploiement sur place pour lequel vous souhaitez spécifier un équilibreur de charge facultatif dans le groupe de déploiement, vous devez avoir créé au moins un Classic Load Balancer, un Application Load Balancer ou un Network Load Balancer dans Elastic Load Balancing. Pour les blue/green déploiements, vous utilisez cet équilibreur de charge pour enregistrer les instances qui constituent votre environnement de remplacement. Les instances de votre environnement original peuvent, le cas échéant, être inscrites auprès de ce même équilibreur de charge. Pour les déploiements sur place, l'équilibreur de charge est utilisé pour désenregistrer les instances sur lesquelles ils travaillent et les réenregistrer une fois le travail terminé. CodeDeploy

CodeDeploy prend en charge blue/green et déploie sur place des instances Amazon EC2 via plusieurs équilibreurs de charge. Supposons, par exemple, que vous disposiez de 200 instances Amazon EC2, dont 100 sont enregistrées auprès de 2 équilibreurs de charge classiques et 100 autres sont enregistrées auprès de 4 groupes cibles dans 2 équilibreurs de charge d'application. Dans ce scénario, vous CodeDeploy pourrez effectuer blue/green et mettre en place des déploiements sur les 200 instances, même s'ils sont répartis entre 2 équilibreurs de charge classiques, 2 équilibreurs de charge d'application et 4 groupes cibles.

CodeDeploy prend en charge jusqu'à 10 équilibreurs de charge classiques et 10 groupes cibles, pour un total de 20 articles.

Pour configurer un ou plusieurs Classic Load Balancers, suivez les instructions de la section [Tutorial : Create a Classic Load](#) Balancer in User Guide for Classic Load Balancers. Notez ce qui suit :

- Dans Étape 2 : Définition d'un équilibreur de charge, dans Créer un équilibreur dans, choisissez le même VPC que vous avez sélectionné lorsque vous avez créé les instances.
- Étape 5 : Inscription des instances EC2 auprès de votre équilibreur de charge, sélectionnez les instances actuellement dans votre groupe de déploiement (déploiements sur-place) ou que vous avez sélectionnées pour être dans votre environnement d'origine (déploiements bleus/verts).
- Dans Étape 7 : Création et vérification de votre équilibreur de charge, notez l'adresse DNS de l'équilibreur de charge.

Par exemple, si vous avez nommé votre équilibreur de charge `my-load-balancer`, votre adresse DNS s'affiche dans un format tel que `my-load-balancer-1234567890.us-east-2.elb.amazonaws.com`.

Pour configurer un ou plusieurs équilibreurs de charge d'application, suivez les instructions de l'une des rubriques suivantes :

- [Création d'un Application Load Balancer](#)
- [Tutoriel : Création d'un Application Load Balancer à l'aide du AWS CLI](#)

Pour configurer un ou plusieurs équilibreurs de charge réseau, suivez les instructions de l'une des rubriques suivantes :

- [Création d'un Network Load Balancer](#)
- [Tutoriel : Création d'un Network Load Balancer à l'aide du AWS CLI](#)

## Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS

Avant d'exécuter un déploiement à l'aide de la plateforme de calcul Amazon ECS, vous devez créer un Application Load Balancer ou un Network Load Balancer, deux groupes cibles et un ou deux écouteurs. Cette rubrique explique comment créer un Application Load Balancer. Pour de plus amples informations, veuillez consulter [Avant de commencer un déploiement Amazon ECS](#).

L'un des groupes cibles dirige le trafic vers l'ensemble de tâches d'origine de votre application Amazon ECS. L'autre groupe cible dirige le trafic vers son ensemble de tâches de remplacement. Au cours du déploiement, CodeDeploy crée un ensemble de tâches de remplacement et redirige le trafic de l'ensemble de tâches d'origine vers le nouveau. CodeDeploy détermine le groupe cible utilisé pour chaque ensemble de tâches.

Un écouteur est utilisé par votre équilibreur de charge pour diriger le trafic vers vos groupes cibles. Un écouteur de production est obligatoire. Vous pouvez spécifier un écouteur de test facultatif qui dirige le trafic vers votre ensemble de tâches de remplacement lors de l'exécution des tests de validation.



L'équilibreur de charge doit utiliser un VPC avec deux sous-réseaux publics dans différentes zones de disponibilité. Les étapes suivantes vous montrent comment confirmer votre VPC par défaut, créer un Amazon EC2 Application Load Balancer, puis créer deux groupes cibles pour votre équilibreur de charge. Pour plus d'informations, consultez la section [Groupes cibles pour vos équilibreurs de charge réseau](#).

## Vérifiez votre VPC, vos sous-réseaux publics et votre groupe de sécurité par défaut

Cette rubrique explique comment créer un Amazon EC2 Application Load Balancer, deux groupes cibles et deux ports pouvant être utilisés lors d'un déploiement Amazon ECS. L'un des ports est facultatif et n'est nécessaire que si vous dirigez le trafic vers un port de test pour les tests de validation lors de votre déploiement.

1. Connectez-vous à la console Amazon VPC AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/vpc/>
2. Vérifiez le VPC par défaut à utiliser. Dans le volet de navigation, sélectionnez Votre VPCs. Notez que le VPC affiche Oui dans la colonne VPC par défaut. Il s'agit de votre VPC par défaut. Il contient les sous-réseaux par défaut que vous utilisez.
3. Choisissez Sous-réseaux. Notez le sous-réseau IDs de deux sous-réseaux qui indiquent Oui dans la colonne Sous-réseau par défaut. Vous les utilisez IDs lorsque vous créez votre équilibreur de charge.
4. Choisissez chaque sous-réseau, puis choisissez l'onglet Description. Vérifiez que les sous-réseaux que vous souhaitez utiliser sont dans des zones de disponibilité différentes.
5. Choisissez les sous-réseaux, puis choisissez l'onglet Table de routage. Pour vérifier que chaque sous-réseau que vous souhaitez utiliser est un sous-réseau public, vérifiez qu'une ligne comportant un lien vers une passerelle Internet est incluse dans la table de routage.
6. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/ec2/>
7. Depuis le panneau de navigation, sélectionnez Groupes de sécurité.
8. Vérifiez que le groupe de sécurité que vous souhaitez utiliser est disponible et notez son ID de groupe (par exemple, sg-abcd1234). Vous l'utilisez lors de la création de votre équilibreur de charge.

## Création d'un Amazon EC2 Application Load Balancer, de deux groupes cibles et d'écouteurs (console)

Pour utiliser la console Amazon EC2 afin de créer un Amazon EC2 Application Load Balancer :

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse.  
<https://console.aws.amazon.com/ec2/>
2. Dans le volet de navigation, choisissez Load Balancers.
3. Sélectionnez Create Load Balancer (Créer un équilibreur de charge).
4. Choisissez Équilibreur de charge d'application, puis choisissez Créer.
5. Dans Nom, entrez le nom de votre équilibreur de charge.
6. Dans Méthode, choisissez Accessible sur Internet.
7. Dans Type d'adresse IP, choisissez ipv4.
8. (Facultatif) Configurez un deuxième port d'écoute pour votre équilibreur de charge. Vous pouvez exécuter des tests de validation de déploiement à l'aide du trafic de test qui est diffusé vers ce port.
  - a. Sous Protocole de l'équilibreur de charge, choisissez Ajouter un écouteur.
  - b. Sous Protocole de l'équilibreur de charge pour le deuxième écouteur, choisissez HTTP.
  - c. Sous Port de l'équilibreur de charge, entrez **8080**.
9. Sous Zones de disponibilité, dans VPC, choisissez le VPC par défaut, puis sélectionnez les deux sous-réseaux par défaut que vous souhaitez utiliser.
10. Choisissez Next: Configure Security Settings (Suivant : Configurer les paramètres de sécurité).
11. Choisissez Next: Configure Security Groups (Suivant : Configurer des groupes de sécurité).
12. Choisissez Select an existing security group (Sélectionner un groupe de sécurité existant), choisissez le groupe de sécurité par défaut, puis notez son ID.
13. Choisissez Next: Configure Routing (Suivant : Configurer le routage).
14. Dans Groupe cible, choisissez Nouveau groupe cible et configurez votre premier groupe cible :
  - a. Dans Nom, entrez un nom de groupe cible (par exemple, **target-group-1**).
  - b. Dans Type cible, choisissez IP.
  - c. Dans Protocole, choisissez HTTP. Dans Port, entrez **80**.
  - d. Choisissez Next: Register Targets (Suivant : Enregistrer des cibles).
15. Choisissez Suivant : Vérification, puis Créer.

Pour créer un deuxième groupe cible pour votre équilibreur de charge

1. Une fois votre équilibreur de charge configuré, ouvrez la console Amazon EC2. Dans le volet de navigation, sélectionnez Groupes cibles.
2. Sélectionnez Créer un groupe cible.
3. Dans Nom, entrez un nom de groupe cible (par exemple, **target-group-2**).
4. Dans Type cible, choisissez IP.
5. Dans Protocole, choisissez HTTP. Dans Port, entrez **80**.
6. Dans VPC, choisissez le VPC par défaut.
7. Choisissez Créer.

 Note

Vous devez avoir créé deux groupes cibles pour votre équilibreur de charge afin que votre déploiement Amazon ECS puisse s'exécuter. Vous utilisez l'ARN de l'un de vos groupes cibles lorsque vous créez votre service Amazon ECS. Pour plus d'informations, consultez [Étape 4 : Création d'un service Amazon ECS](#) dans le guide de l'utilisateur Amazon ECS.

## Création d'un Amazon EC2 Application Load Balancer, de deux groupes cibles et d'écouteurs (CLI)

Pour créer un Application Load Balancer à l'aide de : AWS CLI

1. Utilisez la [create-load-balancer](#) commande pour créer un Application Load Balancer. Spécifiez deux sous-réseaux qui ne se trouvent pas dans la même zone de disponibilité et un groupe de sécurité.

```
aws elbv2 create-load-balancer --name bluegreen-alb \
--subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
region us-east-1
```

Les données de sortie contiennent l'Amazon Resource Name (ARN) de l'équilibreur de charge, au format suivant :

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642
```

2. Utilisez la [create-target-group](#) commande pour créer votre premier groupe cible. CodeDeploy achemine le trafic de ce groupe cible vers la tâche d'origine ou la tâche de remplacement définie dans votre service.

```
aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

Les données de sortie contiennent l'ARN du premier groupe cible, au format suivant :

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4
```

3. Utilisez la [create-target-group](#) commande pour créer votre deuxième groupe cible. CodeDeploy achemine le trafic du groupe cible vers l'ensemble de tâches qui n'est pas desservi par votre premier groupe cible. Par exemple, si votre premier groupe cible achemine le trafic vers l'ensemble de tâches d'origine, ce groupe cible achemine le trafic vers l'ensemble de tâches de remplacement.

```
aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

Les données de sortie contiennent l'ARN du deuxième groupe cible, au format suivant :

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Utilisez la commande [create-listener](#) pour créer un écouteur avec une règle par défaut qui transfère le trafic de production vers le port 80.

```
aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 80 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

Les données de sortie contiennent l'ARN de l'écouteur, au format suivant :

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. (Facultatif) Utilisez la commande [create-listener](#) pour créer un deuxième écouteur avec une règle par défaut qui transfère le trafic de test vers le port 8080. Vous pouvez exécuter des tests de validation de déploiement à l'aide du trafic de test qui est diffusé vers ce port.

```
aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 8080 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4 --region us-east-1
```

Les données de sortie contiennent l'ARN de l'écouteur, au format suivant :

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

## Création d'un groupe de déploiement (CLI)

Pour utiliser le AWS CLI pour créer un groupe de déploiement, appelez la [create-deployment-group](#) commande en spécifiant :

- Le nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Nom pour le groupe de déploiement. Un groupe de déploiement portant ce nom est créé pour l'application spécifiée. Un groupe de déploiement ne peut être associé qu'à une seule application.
- Informations sur les balises, les groupes de balises ou les noms de groupes Amazon EC2 Auto Scaling qui identifient les instances à inclure dans le groupe de déploiement.
- L'identifiant Amazon Resource Name (ARN) du rôle de service qui permet d'agir CodeDeploy au nom de votre AWS compte lorsque vous interagissez avec d'autres AWS services. Pour obtenir le nom ARN du rôle de service, consultez la rubrique [Obtenez l'ARN du rôle de service \(CLI\)](#). Pour

plus d'informations sur les rôles de service, consultez la section [Termes et concepts relatifs aux rôles](#) dans le Guide de l'utilisateur d'IAM.

- Les informations sur le type de déploiement, sur place ou bleu/vert, à associer au groupe de déploiement.
- (Facultatif) Le nom d'une configuration de déploiement existante. Pour afficher la liste des configurations de déploiement, consultez [View Deployment Configuration Details](#). Si aucun nom n'est spécifié, CodeDeploy utilise une configuration de déploiement par défaut.
- (Facultatif) Commandes permettant de créer un déclencheur qui envoie des notifications concernant les événements de déploiement et d'instance aux personnes abonnées à une rubrique Amazon Simple Notification Service. Pour de plus amples informations, veuillez consulter [Monitoring Deployments with Amazon SNS Event Notifications](#).
- (Facultatif) Commandes permettant d'ajouter des CloudWatch alarmes existantes au groupe de déploiement qui sont activées si une métrique spécifiée dans une alarme tombe en dessous ou dépasse un seuil défini.
- (Facultatif) Commandes permettant à un déploiement de revenir à la dernière version correcte connue en cas d'échec du déploiement ou d'activation CloudWatch d'une alarme.
- (Facultatif) Commandes pour un déploiement afin de générer des hooks d'événements du cycle de vie lors d'un événement de scale-in Auto Scaling. Pour de plus amples informations, veuillez consulter [Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy](#).
- Pour les déploiements sur place :
  - (Facultatif) Les noms des équilibres de charge classiques, des équilibres de charge d'application ou des équilibres de charge réseau dans Elastic Load Balancing qui gèrent le trafic vers les instances pendant les processus de déploiement.
- Pour les blue/green déploiements :
  - Configuration du processus blue/green de déploiement :
    - Comment les nouvelles instances dans l'environnement de remplacement sont mises en service.
    - S'il faut réacheminer immédiatement le trafic vers l'environnement de remplacement ou attendre une période donnée pour que le trafic soit réacheminé manuellement.
    - Si les instances dans l'environnement d'origine doivent être résiliées.
  - Les noms des équilibres de charge classiques, des équilibres de charge d'application ou des équilibres de charge réseau dans Elastic Load Balancing à utiliser pour les instances enregistrées dans l'environnement de remplacement.

 Warning

Si vous configurez à la fois un groupe Auto Scaling et un équilibreur de charge Elastic Load Balancing dans votre groupe de déploiement et que vous souhaitez [associer l'équilibreur de charge au groupe Auto Scaling](#), nous vous recommandons de compléter cette pièce jointe avant de créer le CodeDeploy déploiement à partir de ce groupe de déploiement. Toute tentative de terminer la pièce jointe après avoir créé le déploiement peut entraîner le désenregistrement inattendu de toutes les instances de l'équilibreur de charge.

## Afficher les détails du groupe de déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails de tous les groupes de déploiement associés à une application.

### Rubriques

- [Afficher les détails du groupe de déploiement \(console\)](#)
- [Afficher les détails du groupe de déploiement \(CLI\)](#)

## Afficher les détails du groupe de déploiement (console)

Pour utiliser la CodeDeploy console afin de consulter les détails du groupe de déploiement, procédez comme suit :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Sur la page Applications, choisissez le nom de l'application associé au groupe de déploiement.

 Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy n'est pris en charge que dans ces régions.

4. Pour afficher les détails sur un groupe de déploiement individuel, sur l'onglet Groupes de déploiement, choisissez le nom du groupe de déploiement.

## Afficher les détails du groupe de déploiement (CLI)

Pour utiliser le AWS CLI pour afficher les détails du groupe de déploiement, appelez la `get-deployment-group` commande ou la `list-deployment-groups` commande.

Pour afficher des détails sur un groupe de déploiement unique, appelez la commande [get-deployment-group](#), en spécifiant :

- Le nom de l'application associée au groupe de déploiement. Pour obtenir le nom de l'application, appelez la commande [list-applications](#).
- Le nom du groupe de déploiement. Pour obtenir le nom du groupe de déploiement, appelez la commande [list-deployment-groups](#).

Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#), en spécifiant le nom de l'application associée aux groupes de déploiement. Pour obtenir le nom de l'application, appelez la commande [list-applications](#).

## Modifiez les paramètres du groupe de déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour modifier les paramètres d'un groupe de déploiement.



 Warning

N'utilisez pas ces étapes si vous souhaitez que le groupe de déploiement utilise un groupe de déploiement not-yet-created personnalisé. Au lieu de cela, suivez les instructions de la rubrique [Create a Deployment Configuration](#), puis revenez à cette rubrique. N'utilisez pas ces étapes si vous souhaitez que le groupe de déploiement utilise un autre rôle not-yet-created de service. Le rôle de service doit CodeDeploy disposer, au minimum, des autorisations décrites dans [Étape 2 : créer un rôle de service pour CodeDeploy](#). Pour créer et configurer un rôle de service avec les autorisations appropriées, suivez les instructions de l'[Étape 2 : créer un rôle de service pour CodeDeploy](#), puis revenez à cette rubrique.

## Rubriques

- [Modifier les paramètres du groupe de déploiement \(console\)](#)
- [Modifier les paramètres du groupe de déploiement \(CLI\)](#)

## Modifier les paramètres du groupe de déploiement (console)

Pour utiliser la CodeDeploy console afin de modifier les paramètres du groupe de déploiement :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Dans la liste des applications, choisissez le nom de l'application qui est associée au groupe de déploiement que vous voulez modifier.

 Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions

répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy est pris en charge dans ces régions uniquement.

4. Choisissez l'onglet Groupes de déploiement, puis choisissez le nom du groupe de déploiement que vous souhaitez modifier.
5. Sur la page Groupe de déploiement, choisissez Modifier.
6. Modifiez les options du groupe de déploiement si nécessaire.

Pour plus d'informations sur les composants de groupe de déploiement, consultez [Créez un groupe de déploiement avec CodeDeploy](#).

7. Sélectionnez Enregistrer les modifications.

## Modifier les paramètres du groupe de déploiement (CLI)

Pour utiliser le AWS CLI pour modifier les paramètres du groupe de déploiement, appelez la [update-deployment-group](#) commande en spécifiant :

- Pour les déploiements EC2/sur site et Lambda AWS :
  - Le nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
  - Le nom de groupe de déploiement actuel. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).
  - (Facultatif) Un autre nom de groupe de déploiement.
  - (Facultatif) Un autre nom de ressource Amazon (ARN) correspondant à un rôle de service qui permet d'agir CodeDeploy au nom de votre AWS compte lorsque vous interagissez avec d'autres AWS services. Pour obtenir le nom ARN du rôle de service, consultez la rubrique [Obtenez l'ARN du rôle de service \(CLI\)](#). Pour plus d'informations sur les rôles de service, consultez la section [Termes et concepts relatifs aux rôles](#) dans le Guide de l'utilisateur d'IAM.
  - (Facultatif) Le nom de la configuration de déploiement. Pour afficher la liste des configurations de déploiement, consultez [View Deployment Configuration Details](#). (Si ce n'est pas spécifié, CodeDeploy utilise une configuration de déploiement par défaut.)
  - (Facultatif) Commandes permettant d'ajouter une ou plusieurs CloudWatch alarmes existantes au groupe de déploiement qui sont activées si une métrique spécifiée dans une alarme tombe en dessous ou dépasse un seuil défini.

- (Facultatif) Commandes permettant à un déploiement de revenir à la dernière version correcte connue en cas d'échec du déploiement ou d'activation CloudWatch d'une alarme.
- (Facultatif) Commandes pour un déploiement afin de générer des hooks d'événements du cycle de vie lors d'un événement de scale-in Auto Scaling. Pour de plus amples informations, veuillez consulter [Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy](#).
- (Facultatif) Commandes permettant de créer ou de mettre à jour un déclencheur qui publie dans une rubrique d'Amazon Simple Notification Service, afin que les abonnés à cette rubrique reçoivent des notifications concernant les événements de déploiement et d'instance dans ce groupe de déploiement. Pour plus d'informations, consultez [Monitoring Deployments with Amazon SNS Event Notifications](#).
- Pour les déploiements EC2/sur site uniquement :
  - (Facultatif) Des balises de remplacement ou groupes de balises qui identifient de façon unique les instances à inclure dans le groupe de déploiement.
  - (Facultatif) Les noms des groupes Amazon EC2 Auto Scaling de remplacement à ajouter au groupe de déploiement.
- Pour les déploiements Amazon ECS uniquement :
  - Le service Amazon ECS à déployer.
  - Informations sur l'équilibreur de charge, notamment l'Application Load Balancer ou le Network Load Balancer, les groupes cibles requis pour le déploiement d'Amazon ECS, ainsi que les informations relatives à la production et aux auditeurs de test facultatifs.

## Configuration des options avancées d'un groupe de déploiement

Lorsque vous créez ou mettez à jour un groupe de déploiement, vous pouvez configurer un certain nombre d'options pour fournir plus de contrôle et de surveillance sur les déploiements pour ce groupe de déploiement.

Utilisez les informations sur cette page pour vous aider à configurer des options avancées lorsque vous employez des groupes de déploiement dans les rubriques suivantes :

- [Créez une application avec CodeDeploy](#)
- [Créez un groupe de déploiement avec CodeDeploy](#)
- [Modifiez les paramètres du groupe de déploiement avec CodeDeploy](#)

Déclencheurs de notification Amazon SNS : vous pouvez ajouter des déclencheurs à un groupe de CodeDeploy déploiement pour recevoir des notifications concernant les événements liés aux déploiements dans ce groupe de déploiement. Ces notifications sont envoyées aux destinataires abonnés à une rubrique Amazon SNS que vous avez intégrée à l'action du déclencheur.

Vous devez déjà avoir configuré la rubrique Amazon SNS vers laquelle ce déclencheur pointera, et vous CodeDeploy devez être autorisé à publier sur cette rubrique à partir de ce groupe de déploiement. Si vous n'avez pas encore terminé ces étapes de configuration, vous pouvez ajouter ultérieurement des déclencheurs au groupe de déploiement.

Si vous souhaitez créer un déclencheur pour recevoir des notifications sur les événements de déploiement et d'instance dans le groupe de déploiement pour cette application, choisissez Create trigger.

Si votre déploiement concerne une instance Amazon EC2, vous pouvez créer des notifications pour les instances et en recevoir à leur sujet.

Pour de plus amples informations, veuillez consulter [Monitoring Deployments with Amazon SNS Event Notifications](#).

CloudWatch Alarmes Amazon : vous pouvez créer une CloudWatch alarme qui surveille une seule métrique sur une période que vous spécifiez et exécute une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. Pour un déploiement Amazon EC2, vous pouvez créer une alarme pour une instance ou un groupe Amazon EC2 Auto Scaling que vous utilisez dans le cadre de vos opérations. CodeDeploy Pour un déploiement AWS Lambda et Amazon ECS, vous pouvez créer une alarme en cas d'erreur dans une fonction Lambda.

Vous pouvez configurer un déploiement pour qu'il s'arrête lorsqu'une CloudWatch alarme Amazon détecte qu'une métrique est tombée en dessous ou dépasse un seuil défini.

Vous devez déjà avoir créé l'alarme CloudWatch avant de pouvoir l'ajouter à un groupe de déploiement.

1. Pour ajouter la surveillance d'alarmes au groupe de déploiement, dans Alarmes, choisissez Ajouter une alarme.
2. Entrez le nom d'une CloudWatch alarme que vous avez déjà configurée pour surveiller ce déploiement.

Vous devez saisir l' CloudWatch alarme exactement telle qu'elle a été créée CloudWatch. Pour afficher la liste des alarmes, ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>, puis choisissez ALARM.

Options supplémentaires :

- Si vous souhaitez que les déploiements continuent sans tenir compte des alarmes vous avez ajoutées, choisissez Ignore alarm configuration.

Ce choix s'avère utile lorsque vous voulez désactiver temporairement la surveillance des alarmes d'un groupe de déploiement sans devoir ajouter des mêmes alarmes plus tard.

- (Facultatif) Si vous souhaitez que les déploiements se poursuivent en cas d'impossibilité de récupérer l'état de l'alarme auprès d'Amazon CloudWatch, choisissez Poursuivre les déploiements même si le statut de l'alarme n'est pas disponible. CodeDeploy

 Note

Cette option correspond ignorePollAlarmFailure à l'[AlarmConfiguration](#) objet de l' CodeDeploy API.

Pour de plus amples informations, veuillez consulter [Surveillance des déploiements avec des CloudWatch alarmes dans CodeDeploy](#).

**Automatic rollbacks :** Vous pouvez configurer un groupe de déploiement ou un déploiement pour être restauré automatiquement lorsqu'un déploiement échoue, ou lorsqu'un seuil de surveillance que vous spécifiez est atteint. Dans ce cas, la dernière version correcte connue d'une révision de l'application est déployée. Vous pouvez configurer des paramètres facultatifs pour un groupe de déploiement lorsque vous utilisez la console pour créer une application, créer un groupe de déploiement ou mettre à jour un groupe de déploiement. Lorsque vous créez un nouveau déploiement, vous pouvez également choisir de remplacer la configuration de la restauration automatique qui a été spécifiée pour le groupe de déploiement.

- Vous pouvez activer la restauration par les déploiements de la plus récente révision correcte connue en cas de problème en choisissant l'un et/ou l'autre des éléments suivants :

- Rétrogradez en cas d'échec d'un déploiement. CodeDeploy redéploiera la dernière bonne révision connue en tant que nouveau déploiement.
- Roll back when alarm thresholds are met. Si vous avez ajouté une alarme à cette application à l'étape précédente, la dernière version correcte connue CodeDeploy sera redéployée lorsqu'une ou plusieurs des alarmes spécifiées seront activées.

 Note

Pour ignorer temporairement une configuration de restauration, choisissez Disable rollbacks. Ce choix s'avère utile lorsque vous voulez désactiver temporairement des restaurations automatiques sans avoir besoin de mettre en place la même configuration ultérieurement.

Pour de plus amples informations, veuillez consulter [Redéployez et annulez un déploiement avec CodeDeploy](#).

Mises à jour automatiques des instances obsolètes : dans certaines circonstances, vous CodeDeploy pouvez déployer une version obsolète de votre application sur vos instances Amazon EC2. Par exemple, si vos instances EC2 sont lancées dans un groupe Auto Scaling (ASG) alors qu'un CodeDeploy déploiement est en cours, ces instances reçoivent l'ancienne version de votre application au lieu de la dernière. Pour mettre ces instances à jour, lance CodeDeploy automatiquement un déploiement de suivi (immédiatement après le premier) afin de mettre à jour les instances obsolètes. Si vous souhaitez modifier ce comportement par défaut afin de conserver les instances EC2 obsolètes dans l'ancienne version, vous pouvez le faire via l' CodeDeploy API ou la AWS Command Line Interface (CLI).

Pour configurer les mises à jour automatiques des instances obsolètes via l'API, incluez le paramètre de outdatedInstancesStrategy requête dans l>CreateDeploymentGroup action UpdateDeploymentGroup ou. Pour plus de détails, consultez la référence de AWS CodeDeploy l'API.

Pour configurer les mises à jour automatiques via le AWS CLI, utilisez l'une des commandes suivantes :

```
aws deploy update-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

Ou...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

... où *arguments* est remplacé par les arguments requis pour votre déploiement, et *UPDATE|IGNORE* est remplacé par soit pour activer les mises UPDATE à jour automatiques, soit IGNORE pour les désactiver.

Exemple :

```
aws deploy update-deployment-group --application-name "MyApp" --current-
deployment-group-name "MyDG" --region us-east-1 --outdated-instances-
strategy IGNORE
```

Pour plus de détails sur ces AWS CLI commandes, consultez la référence des AWS CLI commandes.

## Supprimer un groupe de déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour supprimer les groupes de déploiement associés à votre AWS compte.

### Warning

Si vous supprimez un groupe de déploiement, tous les détails associés à ce groupe de déploiement seront également supprimés CodeDeploy. Les instances utilisées dans le groupe de déploiement restent inchangées. Cette action ne peut pas être annulée.

### Rubriques

- [Supprimer un groupe de déploiement \(console\)](#)
- [Supprimer un groupe de déploiement \(CLI\)](#)

## Supprimer un groupe de déploiement (console)

Pour utiliser la CodeDeploy console afin de supprimer un groupe de déploiement :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Dans la liste des applications, choisissez le nom de l'application associée au groupe de déploiement.
4. Sur la page Détails de l'application, sur l'onglet Groupes de déploiement, choisissez le nom du groupe de déploiement que vous souhaitez supprimer.
5. Sur la page Détails du déploiement, choisissez Supprimer.
6. A l'invite, tapez le nom du groupe de déploiement pour confirmer que vous souhaitez le supprimer, puis choisissez Supprimer.

## Supprimer un groupe de déploiement (CLI)

AWS CLI Pour supprimer un groupe de déploiement, appelez la [delete-deployment-group](#) commande en spécifiant :

- Le nom de l'application associée au groupe de déploiement. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Le nom du groupe de déploiement associé à l'application. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).



# Travailler avec les révisions d'applications pour CodeDeploy

Dans CodeDeploy, une révision contient une version des fichiers source qui CodeDeploy sera déployée sur vos instances ou des scripts CodeDeploy qui s'exécuteront sur vos instances.

Vous planifiez la révision, ajoutez un AppSpec fichier à la révision, puis vous transférez la révision vers Amazon S3 ou GitHub. Une fois la révision transmise en mode push, vous pouvez la déployer.

## Rubriques

- [Planifier une révision pour CodeDeploy](#)
- [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#)
- [Choisissez un type de CodeDeploy référentiel](#)
- [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#)
- [Consultez les détails de révision de l'application avec CodeDeploy](#)
- [Enregistrez une révision d'application dans Amazon S3 avec CodeDeploy](#)

## Planifier une révision pour CodeDeploy

Une bonne planification facilite grandement le déploiement des révisions.

Pour les déploiements sur une plateforme de calcul AWS Lambda ou Amazon ECS, une révision est identique AppSpec au fichier. Les informations suivantes ne s'appliquent pas. Pour plus d'informations, consultez [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

Pour les déploiements sur une plate-forme de calcul EC2 /On-Premises, commencez par créer un répertoire racine (dossier) vide sur la machine de développement. C'est là que vous stockerez les fichiers sources (tels que les fichiers texte et binaires, les fichiers exécutables, les packages, etc.) à déployer sur les instances ou les scripts à exécuter sur les instances.

Par exemple, dans le dossier /tmp/ racine sous Linux, macOS ou Unix ou dans le dossier c:\temp racine sous Windows :

```
/tmp/ or c:\temp (root folder)
|--content (subfolder)
```

```
| |--myTextFile.txt
| |--mySourceFile.rb
| |--myExecutableFile.exe
| |--myInstallerFile.msi
| |--myPackage.rpm
| |--myImageFile.png
|--scripts (subfolder)
| |--myShellScript.sh
| |--myBatchScript.bat
| |--myPowerShellScript.ps1
|--appspec.yml
```

Le dossier racine doit également inclure un fichier de spécification de l'application (AppSpec fichier), comme illustré ici. Pour de plus amples informations, veuillez consulter [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

## Ajouter un fichier de spécification d'application à une révision pour CodeDeploy

Cette rubrique explique comment ajouter un AppSpec fichier à votre déploiement. Il inclut également des modèles permettant de créer un AppSpec fichier pour un déploiement AWS Lambda et EC2/on-premises.

### Rubriques

- [Ajouter un AppSpec fichier pour un déploiement Amazon ECS](#)
- [Ajouter un AppSpec fichier pour un déploiement AWS Lambda](#)
- [Ajouter un AppSpec fichier pour un déploiement EC2/sur site](#)

## Ajouter un AppSpec fichier pour un déploiement Amazon ECS

Pour un déploiement sur une plateforme de calcul Amazon ECS :

- Le AppSpec fichier indique la définition de tâche Amazon ECS utilisée pour le déploiement, le nom du conteneur et le mappage des ports utilisés pour acheminer le trafic, ainsi que les fonctions Lambda facultatives exécutées après les événements du cycle de vie du déploiement.
- Une révision est identique à un AppSpec fichier.
- Un AppSpec fichier peut être écrit en utilisant JSON ou YAML.

- Un AppSpec fichier peut être enregistré sous forme de fichier texte ou saisi directement dans une console lorsque vous créez un déploiement. Pour de plus amples informations, veuillez consulter [Création d'un déploiement d'Amazon ECS Compute Platform \(console\)](#).

Pour créer un AppSpec fichier

1. Copiez le modèle JSON ou YAML dans un éditeur de texte ou dans l' AppSpec éditeur de la console.
2. Modifiez le modèle selon les besoins.
3. Utilisez un validateur JSON ou YAML pour valider votre AppSpec fichier. Si vous utilisez l' AppSpec éditeur, le fichier est validé lorsque vous choisissez Créer un déploiement.
4. Enregistrez le fichier si vous utilisez un éditeur de texte. Si vous utilisez le AWS CLI pour créer votre déploiement, référencez le AppSpec fichier s'il se trouve sur votre disque dur ou dans un compartiment Amazon S3. Si vous utilisez la console, vous devez transférer votre AppSpec fichier vers Amazon S3.

## Modèle de AppSpec fichier YAML pour un déploiement Amazon ECS avec instructions

Ce qui suit est un modèle de AppSpec fichier YAML pour un déploiement Amazon ECS avec toutes les options disponibles. Pour plus d'informations sur les événements de cycle de vie à utiliser dans la section hooks, consultez [AppSpec section « hooks » pour un déploiement Amazon ECS](#).

```
This is an appspec.yml template file for use with an Amazon ECS deployment in
 CodeDeploy.
The lines in this template that start with the hashtag are
comments that can be safely left in the file or
ignored.
For help completing this file, see the "AppSpec File Reference" in the
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
In the Resources section, you must specify the following: the Amazon ECS service,
 task definition name,
and the name and port of the load balancer to route traffic,
target version, and (optional) the current version of your AWS Lambda function.
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
```

```

 TaskDefinition: "" # Specify the ARN of your task definition
 (arn:aws:ecs:region:account-id:task-definition/task-definition-family-name:task-
 definition-revision-number)
 LoadBalancerInfo:
 ContainerName: "" # Specify the name of your Amazon ECS application's
 container
 ContainerPort: "" # Specify the port for your container where traffic
 reroutes
Optional properties
 PlatformVersion: "" # Specify the version of your Amazon ECS Service
 NetworkConfiguration:
 AwsvpcConfiguration:
 Subnets: ["", ""] # Specify one or more comma-separated subnets in your
 Amazon ECS service
 SecurityGroups: ["", ""] # Specify one or more comma-separated security
 groups in your Amazon ECS service
 AssignPublicIp: "" # Specify "ENABLED" or "DISABLED"
(Optional) In the Hooks section, specify a validation Lambda function to run during
a lifecycle event.
Hooks:
Hooks for Amazon ECS deployments are:
- BeforeInstall: "" # Specify a Lambda function name or ARN
- AfterInstall: "" # Specify a Lambda function name or ARN
- AfterAllowTestTraffic: "" # Specify a Lambda function name or ARN
- BeforeAllowTraffic: "" # Specify a Lambda function name or ARN
- AfterAllowTraffic: "" # Specify a Lambda function name or ARN

```

## AppSpec Fichier JSON pour un modèle de déploiement Amazon ECS

Ce qui suit est un modèle JSON pour un AppSpec fichier destiné à un déploiement Amazon ECS avec toutes les options disponibles. Pour obtenir des instructions sur les modèles, consultez les commentaires pour la version YAML dans la section précédente. Pour plus d'informations sur les événements de cycle de vie à utiliser dans la section hooks, consultez [AppSpec section « hooks » pour un déploiement Amazon ECS](#).

```

{
 "version": 0.0,
 "Resources": [
 {
 "TargetService": {
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "",

```

```
 "LoadBalancerInfo": {
 "ContainerName": "",
 "ContainerPort":
 },
 "PlatformVersion": "",
 "NetworkConfiguration": {
 "AwsVpcConfiguration": {
 "Subnets": [
 "",
 ""
],
 "SecurityGroups": [
 "",
 ""
],
 "AssignPublicIp": ""
 }
 }
 }
},
"Hooks": [
 {
 "BeforeInstall": ""
 },
 {
 "AfterInstall": ""
 },
 {
 "AfterAllowTestTraffic": ""
 },
 {
 "BeforeAllowTraffic": ""
 },
 {
 "AfterAllowTraffic": ""
 }
]
```

# Ajouter un AppSpec fichier pour un déploiement AWS Lambda

Pour un déploiement sur une plateforme de calcul AWS Lambda :

- Le AppSpec fichier contient des instructions sur les fonctions Lambda à déployer et à utiliser pour la validation du déploiement.
- Une révision est identique à un AppSpec fichier.
- Un AppSpec fichier peut être écrit en utilisant JSON ou YAML.
- Un AppSpec fichier peut être enregistré sous forme de fichier texte ou saisi directement dans un AppSpec éditeur de console lors de la création d'un déploiement. Pour de plus amples informations, veuillez consulter [Création d'un déploiement de plateforme de calcul AWS Lambda \(console\)](#).

Pour créer un AppSpec fichier :

1. Copiez le modèle JSON ou YAML dans un éditeur de texte ou dans l' AppSpec éditeur de la console.
2. Modifiez le modèle selon les besoins.
3. Utilisez un validateur JSON ou YAML pour valider votre AppSpec fichier. Si vous utilisez l' AppSpec éditeur, le fichier est validé lorsque vous choisissez Créer un déploiement.
4. Enregistrez le fichier si vous utilisez un éditeur de texte. Si vous utilisez le AWS CLI pour créer votre déploiement, référencez le AppSpec fichier s'il se trouve sur votre disque dur ou dans un compartiment Amazon S3. Si vous utilisez la console, vous devez transférer votre AppSpec fichier vers Amazon S3.

## Modèle de AppSpec fichier YAML pour un AWS Lambda déploiement avec instructions

Pour plus d'informations sur les événements de cycle de vie à utiliser dans la section hooks, consultez [AppSpec section « hooks » pour un déploiement AWS Lambda](#).

```
This is an appspec.yml template file for use with an AWS Lambda deployment in
CodeDeploy.
The lines in this template starting with the hashtag symbol are
instructional comments and can be safely left in the file or
ignored.
```

```
For help completing this file, see the "AppSpec File Reference" in the
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
In the Resources section specify the name, alias,
target version, and (optional) the current version of your AWS Lambda function.
Resources:
 - MyFunction: # Replace "MyFunction" with the name of your Lambda function
 Type: AWS::Lambda::Function
 Properties:
 Name: "" # Specify the name of your Lambda function
 Alias: "" # Specify the alias for your Lambda function
 CurrentVersion: "" # Specify the current version of your Lambda function
 TargetVersion: "" # Specify the version of your Lambda function to deploy
(Optional) In the Hooks section, specify a validation Lambda function to run during
a lifecycle event. Replace "LifeCycleEvent" with BeforeAllowTraffic
or AfterAllowTraffic.
Hooks:
 - LifeCycleEvent: "" # Specify a Lambda validation function between double-quotes.
```

## AppSpec Fichier JSON pour un modèle AWS Lambda de déploiement

Dans le modèle suivant, remplacez « MyFunction » par le nom de votre AWS Lambda fonction. Dans la section Hooks facultative, remplacez les événements du cycle de vie par BeforeAllowTraffic ou AfterAllowTraffic.

Pour plus d'informations sur les événements de cycle de vie à utiliser dans la section Hooks, consultez [AppSpec section « hooks » pour un déploiement AWS Lambda](#).

```
{
 "version": 0.0,
 "Resources": [{
 "MyFunction": {
 "Type": "AWS::Lambda::Function",
 "Properties": {
 "Name": "",
 "Alias": "",
 "CurrentVersion": "",
 "TargetVersion": ""
 }
 }
]},
 "Hooks": [{
```

```
"LifecycleEvent": ""
}
]
}
```

## Ajouter un AppSpec fichier pour un déploiement EC2/sur site

Sans AppSpec fichier, vous CodeDeploy ne pouvez pas mapper les fichiers source de la révision de votre application à leurs destinations ou exécuter des scripts pour votre déploiement sur une plateforme de calcul EC2/sur site.

Chaque révision ne doit contenir qu'un seul AppSpec fichier.

Pour ajouter un AppSpec fichier à une révision :

1. Copiez le modèle dans un éditeur de texte.
2. Modifiez le modèle selon les besoins.
3. Utilisez un validateur YAML pour vérifier la validité de votre AppSpec fichier.
4. Enregistrez le fichier sous le nom `appspec.yml` dans le répertoire racine de la révision.
5. Exécutez l'une des commandes suivantes pour vérifier que vous avez bien placé votre AppSpec fichier dans le répertoire racine :
  - Pour Linux, macOS ou Unix :

```
find /path/to/root/directory -name appspec.yml
```

Il n'y aura aucune sortie si le AppSpec fichier n'y est pas trouvé.

- Pour Windows :

```
dir path\to\root\directory\appspec.yml
```

Un message d'erreur Fichier introuvable s'affichera si le AppSpec fichier n'y est pas stocké.

6. Transférez la révision vers Amazon S3 ou GitHub.

Pour obtenir des instructions, veuillez consulter [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#).



## AppSpec modèle de fichier pour un déploiement EC2/sur site avec instructions

### Note

Les déploiements sur des instances Windows Server ne prennent pas en charge cet élément. Si vous effectuez un déploiement sur des instances Windows Server, ne l'incluez pas dans votre AppSpec fichier.

```
This is an appspec.yml template file for use with an EC2/On-Premises deployment in
CodeDeploy.
The lines in this template starting with the hashtag symbol are
instructional comments and can be safely left in the file or
ignored.
For help completing this file, see the "AppSpec File Reference" in the
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
Specify "os: linux" if this revision targets Amazon Linux,
Red Hat Enterprise Linux (RHEL), or Ubuntu Server
instances.
Specify "os: windows" if this revision targets Windows Server instances.
(You cannot specify both "os: linux" and "os: windows".)
os: linux
os: windows
During the Install deployment lifecycle event (which occurs between the
BeforeInstall and AfterInstall events), copy the specified files
in "source" starting from the root of the revision's file bundle
to "destination" on the Amazon EC2 instance.
Specify multiple "source" and "destination" pairs if you want to copy
from multiple sources or to multiple destinations.
If you are not copying any files to the Amazon EC2 instance, then remove the
"files" section altogether. A blank or incomplete "files" section
may cause associated deployments to fail.
files:
 - source:
 destination:
 - source:
 destination:
For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
you can specify a "permissions"
section here that describes special permissions to apply to the files
```

```
in the "files" section as they are being copied over to
the Amazon EC2 instance.
For more information, see the documentation.
If you are deploying to Windows Server instances,
then remove the
"permissions" section altogether. A blank or incomplete "permissions"
section may cause associated deployments to fail.
permissions:
 - object:
 pattern:
 except:
 owner:
 group:
 mode:
 acls:
 -
 context:
 user:
 type:
 range:
 type:
 -

If you are not running any commands on the Amazon EC2 instance, then remove
the "hooks" section altogether. A blank or incomplete "hooks" section
may cause associated deployments to fail.
hooks:
For each deployment lifecycle event, specify multiple "location" entries
if you want to run multiple scripts during that event.
You can specify "timeout" as the number of seconds to wait until failing the
deployment
if the specified scripts do not run within the specified time limit for the
specified event. For example, 900 seconds is 15 minutes. If not specified,
the default is 1800 seconds (30 minutes).
Note that the maximum amount of time that all scripts must finish executing
for each individual deployment lifecycle event is 3600 seconds (1 hour).
Otherwise, the deployment will stop and CodeDeploy will consider the deployment
to have failed to the Amazon EC2 instance. Make sure that the total number of
seconds
that are specified in "timeout" for all scripts in each individual deployment
lifecycle event does not exceed a combined 3600 seconds (1 hour).
For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
you can specify "runas" in an event to
run as the specified user. For more information, see the documentation.
If you are deploying to Windows Server instances,
```

```
remove "runas" altogether.
If you do not want to run any commands during a particular deployment
lifecycle event, remove that event declaration altogether. Blank or
incomplete event declarations may cause associated deployments to fail.
During the ApplicationStop deployment lifecycle event, run the commands
in the script specified in "location" starting from the root of the
revision's file bundle.
ApplicationStop:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the BeforeInstall deployment lifecycle event, run the commands
in the script specified in "location".
BeforeInstall:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the AfterInstall deployment lifecycle event, run the commands
in the script specified in "location".
AfterInstall:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the ApplicationStart deployment lifecycle event, run the commands
in the script specified in "location".
ApplicationStart:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the ValidateService deployment lifecycle event, run the commands
in the script specified in "location".
ValidateService:
```

```
- location:
 timeout:
 runas:
- location:
 timeout:
 runas:
```

## Choisissez un type de CodeDeploy référentiel

L'emplacement de stockage des fichiers requis par CodeDeploy est appelé référentiel. L'utilisation d'un référentiel dépend de la plate-forme informatique utilisée par votre déploiement.

- **EC2/On-Premises** : pour déployer le code de votre application sur une ou plusieurs instances, votre code doit être regroupé dans un fichier d'archive et placé dans un référentiel auquel CodeDeploy vous pouvez accéder pendant le processus de déploiement. Vous regroupez votre contenu déployable et un AppSpec fichier dans un fichier d'archive, puis vous le chargez dans l'un des types de référentiels pris en charge par CodeDeploy.
- **AWS Lambda et Amazon ECS** : les déploiements nécessitent un AppSpec fichier auquel il est possible d'accéder au cours d'un déploiement de l'une des manières suivantes :
  - À partir d'un compartiment Amazon S3.
  - À partir du texte saisi directement dans l' AppSpec éditeur de la console. Pour plus d'informations, consultez [Création d'un déploiement de plateforme de calcul AWS Lambda \(console\)](#) et [Création d'un déploiement d'Amazon ECS Compute Platform \(console\)](#).
  - Si vous utilisez le AWS CLI, vous pouvez référencer un AppSpec fichier qui se trouve sur votre disque dur ou sur un lecteur réseau. Pour plus d'informations, consultez [Création d'un déploiement de plateforme de calcul AWS Lambda \(CLI\)](#) et [Création d'un déploiement de plateforme de calcul Amazon ECS \(CLI\)](#).

CodeDeploy prend actuellement en charge les types de référentiels suivants :

| Type de référentiel | Détails de référentiel                                                                                              | Plate-forme de calcul prise en charge                                                 |
|---------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Amazon S3           | <a href="#">Amazon Simple Storage Service</a> (Amazon S3) est la AWS solution pour le stockage d'objets sécurisé et | Les déploiements qui utilisent les plateformes de calcul suivantes peuvent stocker la |

évolutif. Amazon S3 stocke les données sous forme d'objets dans des compartiments. Un objet se compose d'un fichier et, éventuellement, de métadonnées décrivant ce fichier.

Pour stocker un objet dans Amazon S3, vous devez charger le fichier dans un compartiment. Lorsque vous chargez un fichier, vous pouvez définir des autorisations et des métadonnées sur cet objet.

En savoir plus :

- [Création d'un compartiment dans Amazon S3](#)
- [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#)
- [Déployez automatiquement depuis Amazon S3 à l'aide de CodeDeploy](#)

révision dans un compartiment Amazon S3.

- EC2/Sur site
- AWS Lambda
- Amazon ECS

## GitHub

Vous pouvez stocker les révisions de votre application dans des [GitHub](#) référentiels. Vous pouvez déclencher un déploiement à partir d'un GitHub référentiel chaque fois que le code source de ce référentiel est modifié.

En savoir plus :

- [Intégration CodeDeploy avec GitHub](#)
- [Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub](#)

Seuls les EC2 déploiements / On-Premises peuvent stocker la révision dans un référentiel. GitHub

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                     |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Bitbucket | <p>Vous pouvez déployer du code sur des groupes d' EC2 instances de déploiement en utilisant le <a href="#">CodeDeploy canal dans Bitbucket Pipelines</a> . Bitbucket Pipelines offre des fonctionnalités d'intégration et de déploiement en continu (CI/ CD), y compris des <a href="#">déploiements Bitbucket</a>. Le CodeDeploy canal envoie d'abord l'artefact vers un compartiment S3 que vous avez spécifié, puis déploie l'artefact de code depuis le compartiment.</p> <p>En savoir plus :</p> <ul style="list-style-type: none"><li>• <a href="#">Découvrez le CodeDeploy tube de Bitbucket</a></li></ul> | Seuls les EC2 déploiements / On-Premises peuvent stocker la révision dans un référentiel. BitBucket |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|

 Note

Un AWS Lambda déploiement fonctionne uniquement avec un référentiel Amazon S3.

## Transférer une révision CodeDeploy pour Amazon S3 (déploiements EC2/sur site uniquement)

Après avoir planifié votre révision comme décrit dans [Planifier une révision pour CodeDeploy](#) et ajouté un AppSpec fichier à la révision comme décrit dans [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#), vous êtes prêt à regrouper les fichiers des composants et à transférer la révision vers Amazon S3. Pour les déploiements sur des instances Amazon EC2, une fois que vous avez envoyé la révision, vous pouvez CodeDeploy l'utiliser pour déployer la révision depuis Amazon S3 vers les instances.

 Note

CodeDeploy peut également être utilisé pour déployer des révisions qui ont été transférées vers GitHub. Pour plus d'informations, consultez votre GitHub documentation.

Nous supposons que vous avez déjà suivi les instructions de la rubrique [Commencer avec CodeDeploy](#) pour configurer l'interface AWS CLI. Cela est particulièrement important pour l'appel de la commande push décrite plus tard.

Assurez-vous de disposer d'un compartiment Amazon S3. Suivez les instructions de la [section Créer un bucket](#).

Si votre déploiement concerne des instances Amazon EC2, le compartiment Amazon S3 cible doit être créé ou exister dans la même région que les instances cibles. Par exemple, si vous souhaitez déployer une révision sur certaines instances de la région USA Est (Virginie du Nord) et sur d'autres instances de la région USA Ouest (Oregon), vous devez avoir un bucket dans la région USA Est (Virginie du Nord) avec une copie de la révision et un autre bucket dans la région USA Ouest (Oregon) avec une autre copie de la même révision. Dans ce scénario, vous devrez ensuite créer deux déploiements distincts, l'un dans la région USA Est (Virginie du Nord) et l'autre dans la région USA Ouest (Oregon), même si la révision est la même dans les deux régions et dans les compartiments.

Vous devez disposer des autorisations nécessaires pour effectuer le téléchargement dans le compartiment Amazon S3. Vous pouvez spécifier ces autorisations par le biais d'une politique de compartiment Amazon S3. Par exemple, dans la politique de compartiment Amazon S3 suivante, l'utilisation du caractère générique (\*) permet 111122223333 au AWS compte de télécharger des fichiers dans n'importe quel répertoire du compartiment Amazon S3 nommé amzn-s3-demo-bucket :

```
{
 "Statement": [
 {
 "Action": [
 "s3:PutObject"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "Principal": {
```



```
 "AWS": [
 "111122223333"
]
 }
}
```

Pour consulter l'identifiant de votre AWS compte, consultez la section [Trouver votre identifiant de AWS compte](#).

Pour savoir comment générer et associer une politique de compartiment Amazon S3, consultez les [exemples de politique de compartiment](#).

L'utilisateur qui appelle la push commande doit disposer, au minimum, des autorisations nécessaires pour télécharger la révision dans chaque compartiment Amazon S3 cible. Par exemple, la politique suivante permet à l'utilisateur de télécharger des révisions n'importe où dans le compartiment Amazon S3 nommé `amzn-s3-demo-bucket` :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
],
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
 }
]
}
```

Pour savoir comment créer et associer une stratégie IAM, consultez la section [Utilisation des politiques](#).

## Envoyez une révision à l'aide du AWS CLI

### Note

La push commande regroupe les artefacts de l'application et un AppSpec fichier dans une révision. Le format de fichier de cette révision est compressé dans un fichier ZIP. La commande ne peut pas être utilisée avec un déploiement AWS Lambda ou Amazon ECS car chacun attend une révision sous forme de fichier au format JSON ou YAML. AppSpec

Appelez la commande push pour grouper et transmettre la révision en mode push pour un déploiement. Ses paramètres sont les suivants :

- `--application-name` : (chaîne) obligatoire. Nom de l'application CodeDeploy à associer à cette révision d'application.
- `--s3-location` : (chaîne) obligatoire. Informations sur l'emplacement de la révision de l'application à télécharger sur Amazon S3. Vous devez spécifier un compartiment Amazon S3 et une clé. La clé est le nom de la révision. CodeDeploy compresse le contenu avant son téléchargement. Utilisez le format `s3://amzn-s3-demo-bucket/your-key.zip`.
- `--ignore-hidden-files` ou `--no-ignore-hidden-files` : (booléen) facultatif. Utilisez `--no-ignore-hidden-files` indicateur (par défaut) pour regrouper et télécharger des fichiers cachés sur Amazon S3. Utilisez `--ignore-hidden-files` indicateur pour ne pas regrouper et télécharger des fichiers cachés sur Amazon S3.
- `--source` (chaîne) facultatif. Emplacement du contenu à déployer et du AppSpec fichier sur la machine de développement à compresser et à télécharger sur Amazon S3. L'emplacement est spécifié en tant que chemin d'accès relatif par rapport au répertoire actuel. Si le chemin d'accès relatif n'est pas spécifié ou si un seul point est utilisé pour le chemin (« . »), le répertoire actuel est utilisé.
- `--description` (chaîne) facultatif. Commentaire qui résume la révision d'application. Si elle n'est pas spécifiée, la chaîne par défaut « Uploaded by AWS CLI 'time' UTC » est utilisée, où « time » est l'heure actuelle du système en temps universel coordonné (UTC).

Vous pouvez utiliser le AWS CLI pour envoyer une révision pour un déploiement Amazon EC2. Un exemple de commande push se présente sous la forme suivante :

Sous Linux, macOS ou Unix :

```
aws deploy push \
 --application-name WordPress_App \
 --description "This is a revision for the application WordPress_App" \
 --ignore-hidden-files \
 --s3-location s3://amzn-s3-demo-bucket/WordPressApp.zip \
 --source .
```

Dans Windows :

```
aws deploy push --application-name WordPress_App --description "This is a revision for
the application WordPress_App" --ignore-hidden-files --s3-location s3://amzn-s3-demo-
bucket/WordPressApp.zip --source .
```

Cette commande exécute les opérations suivantes :

- Elle associe les fichiers groupés à une application nommée WordPress\_App.
- Elle attache une description à la révision.
- Elle ignore les fichiers masqués.
- Elle attribue un nom à la révision WordPressApp.zip et transmet celle-ci en mode push à un compartiment nommé amzn-s3-demo-bucket.
- Elle groupe tous les fichiers dans le répertoire racine dans la révision.

Une fois le push réussi, vous pouvez utiliser la console AWS CLI ou la CodeDeploy console pour déployer la révision depuis Amazon S3. Pour déployer cette révision avec AWS CLI :

Sous Linux, macOS ou Unix :

```
aws deploy create-deployment \
 --application-name WordPress_App \
 --deployment-config-name your-deployment-config-name \
 --deployment-group-name your-deployment-group-name \
 --s3-location bucket=amzn-s3-demo-bucket,key=WordPressApp.zip,bundleType=zip
```

Dans Windows :

```
aws deploy create-deployment --application-name WordPress_App --deployment-config-
name your-deployment-config-name --deployment-group-name your-deployment-group-name --
s3-location bucket=amzn-s3-demo-bucket,key=WordPressApp.zip,bundleType=zip
```

Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).

## Consultez les détails de révision de l'application avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails de toutes les révisions d'applications enregistrées sur votre AWS compte pour une application spécifique.

Pour plus d'informations sur l'inscription d'une révision, consultez la rubrique [Enregistrez une révision d'application dans Amazon S3 avec CodeDeploy](#).

### Rubriques

- [Afficher les détails de révision de l'application \(console\)](#)
- [Afficher les détails de révision de l'application \(CLI\)](#)

## Afficher les détails de révision de l'application (console)

Pour afficher les détails de révision d'application :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

#### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Déployer, puis choisissez Applications.

#### Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy n'est pris en charge que dans ces régions.

3. Choisissez le nom de l'application qui la révision que vous souhaitez afficher.

4. Sur la page Détails de l'application, choisissez l'onglet Révisions et passez en revue la liste des révisions qui sont inscrites pour l'application. Choisissez une révision, puis choisissez Afficher les détails.

## Afficher les détails de révision de l'application (CLI)

AWS CLI Pour afficher la révision d'une application, appelez la `get-application-revision` commande ou la `list-application-revisions` commande.

### Note

Les références GitHub s'appliquent uniquement aux déploiements vers des déploiements EC2/sur site. Les révisions pour les AWS Lambda déploiements ne fonctionnent pas avec GitHub.

Pour afficher des détails sur une révision d'application donnée, appelez la commande [get-application-revision](#), en spécifiant :

- Le nom de l'application. Pour obtenir le nom de l'application, appelez la commande [list-applications](#).
- Pour une révision stockée dans GitHub, le nom du GitHub référentiel et l'ID du commit qui fait référence à la révision de l'application qui a été transférée vers le référentiel.
- Pour une révision stockée dans Amazon S3, le nom du compartiment Amazon S3 contenant la révision ; le nom et le type de fichier du fichier d'archive chargé ; et, éventuellement, l'identifiant de version Amazon S3 du fichier d'archive et ETag. Si l'identifiant de version ETag, ou les deux, ont été spécifiés lors d'un appel à [register-application-revision](#), ils doivent être spécifiés ici.

Pour afficher des détails sur plusieurs révisions d'application, appelez la commande [list-application-revisions](#), en spécifiant :

- Le nom de l'application. Pour obtenir le nom de l'application, appelez la commande [list-applications](#).
- Facultativement, pour afficher les détails des révisions des applications Amazon S3 uniquement, le nom du compartiment Amazon S3 contenant les révisions.

- Facultativement, pour afficher les détails des révisions des applications Amazon S3 uniquement, une chaîne de préfixe pour limiter la recherche aux révisions des applications Amazon S3. (Si ce n'est pas spécifié, CodeDeploy répertoriera toutes les révisions correspondantes de l'application Amazon S3.)
- Eventuellement, s'il convient de répertorier les détails de révision selon que chaque révision est la révision cible d'un groupe de déploiement. (Si ce n'est pas spécifié, CodeDeploy listera toutes les révisions correspondantes.)
- Eventuellement, le nom de colonne et l'ordre selon lesquels il convient de trier la liste des détails de révision. (Si ce n'est pas spécifié, les résultats CodeDeploy seront listés dans un ordre arbitraire.)

Vous pouvez répertorier toutes les révisions ou uniquement les révisions stockées dans Amazon S3. Vous ne pouvez pas répertorier uniquement les révisions stockées dans GitHub.

## Enregistrez une révision d'application dans Amazon S3 avec CodeDeploy

Si vous avez déjà appelé la commande [push](#) pour transférer une révision d'application vers Amazon S3, vous n'avez pas besoin d'enregistrer la révision. Toutefois, si vous chargez une révision sur Amazon S3 par un autre moyen et que vous souhaitez qu'elle apparaisse dans la CodeDeploy console ou via le AWS CLI, suivez ces étapes pour enregistrer d'abord la révision.

Si vous avez transféré une révision d'application vers un GitHub référentiel et que vous souhaitez qu'elle apparaisse dans la CodeDeploy console ou via le AWS CLI, vous devez également suivre ces étapes.

Vous pouvez uniquement utiliser le AWS CLI ou CodeDeploy APIs pour enregistrer les révisions d'applications dans Amazon S3 ou GitHub.

### Rubriques

- [Enregistrez une révision dans Amazon S3 avec CodeDeploy \(CLI\)](#)
- [Enregistrer une révision GitHub avec CodeDeploy \(CLI\)](#)

## Enregistrez une révision dans Amazon S3 avec CodeDeploy (CLI)

1. Téléchargez la révision sur Amazon S3.

## 2. Appelez la commande [register-application-revision](#), en spécifiant les éléments suivants :

- Le nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Des informations sur la révision à inscrire :
  - Nom du compartiment Amazon S3 qui contient la révision.
  - Le nom et le type de fichier de la révision chargée. Pour les AWS Lambda déploiements, la révision est un AppSpec fichier écrit en JSON ou YAML. Pour les déploiements EC2/sur site, la révision contient une version des fichiers source qui CodeDeploy sera déployée sur vos instances ou des scripts qui s' CodeDeploy exécuteront sur vos instances.

### Note

Les formats de fichier d'archive tar et tar compressé (.tar et .tar.gz) ne sont pas pris en charge pour les instances Windows Server.

- (Facultatif) L'identifiant de version Amazon S3 de la révision. (Si l'identifiant de version n'est pas spécifié, CodeDeploy nous utiliserons la version la plus récente.)
- (Facultatif) La révision ETag. (Si le n' ETag est pas spécifié, la validation de l'objet CodeDeploy sera ignorée.)
- (Facultatif) Toute description que vous souhaitez associer à la révision.

Les informations relatives à une révision dans Amazon S3 peuvent être spécifiées sur la ligne de commande, en utilisant cette syntaxe dans le cadre de l'`register-application-revision` appel. (version et eTag sont facultatifs.)

Pour un fichier de révision pour un déploiement EC2/sur site :

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|zip,version=string,eTag=string
```

Pour un fichier de révision pour un déploiement AWS Lambda :

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

## Enregistrer une révision GitHub avec CodeDeploy (CLI)

### Note

AWS Lambda les déploiements ne fonctionnent pas avec GitHub.

1. Téléchargez la révision dans votre GitHub dépôt.
2. Appelez la commande [register-application-revision](#), en spécifiant les éléments suivants :
  - Le nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
  - Des informations sur la révision à inscrire :
    - Le nom GitHub d'utilisateur ou de groupe attribué au référentiel contenant la révision, suivi d'une barre oblique (/), puis du nom du référentiel.
    - L'ID de la validation qui fait référence à la révision dans le référentiel.
  - (Facultatif) Toute description que vous souhaitez associer à la révision.

Les informations relatives à une révision GitHub peuvent être spécifiées sur la ligne de commande, en utilisant cette syntaxe dans le cadre de l'`register-application-revision`appel :

```
--github-location repository=string,commitId=string
```



# Travailler avec des déploiements dans CodeDeploy

Dans CodeDeploy, un déploiement est le processus, et les composants impliqués dans le processus, d'installation de contenu sur une ou plusieurs instances. Ce contenu peut être constitué de code, de fichiers Web et de configuration, d'exécutables, de packages, de scripts, etc. CodeDeploy déploie le contenu stocké dans un référentiel source, conformément aux règles de configuration que vous spécifiez.

Si vous utilisez la plate-forme de calcul EC2/sur site, deux déploiements sur le même ensemble d'instances peuvent s'exécuter simultanément.

CodeDeploy propose deux types de déploiement, les déploiements sur place et les déploiements bleu/vert.

- **Déploiement sur place** : l'application de chaque instance du groupe de déploiement est arrêtée, la dernière révision de l'application est installée et la nouvelle version de l'application est démarrée et validée. Vous pouvez utiliser un équilibreur de charge afin que chaque instance soit désenregistrée pendant son déploiement, puis remise en service une fois le déploiement terminé. Seuls les déploiements qui utilisent la plate-forme informatique EC2/sur site peuvent utiliser des déploiements sur place. Pour plus d'informations sur les déploiements sur place, consultez [Vue d'ensemble d'un déploiement sur place](#).
- **Déploiement bleu/vert** : le comportement de votre déploiement dépend de la plateforme informatique que vous utilisez :
  - **Blue/green on an EC2/On-Plateforme de calcul sur site** : les instances d'un groupe de déploiement (l'environnement d'origine) sont remplacées par un ensemble d'instances différent (l'environnement de remplacement) en suivant les étapes suivantes :
    - Les instances sont provisionnées pour l'environnement de remplacement.
    - La dernière version de l'application est installée sur les instances de remplacement.
    - Un temps d'attente facultatif se produit pour des activités telles que les tests d'applications et la vérification du système.
    - Les instances de l'environnement de remplacement sont enregistrées auprès d'un ou de plusieurs équilibreurs de charge Elastic Load Balancing, ce qui entraîne le reroutage du trafic vers celles-ci. Les instances de l'environnement d'origine sont désenregistrées et peuvent être résiliées ou maintenues en cours d'exécution pour d'autres utilisations.

 Note

Si vous utilisez une plate-forme de calcul EC2/sur site, sachez que les blue/green déploiements fonctionnent uniquement avec les instances Amazon EC2.

- Bleu/vert sur une plateforme de calcul AWS Lambda Amazon ECS ou Amazon ECS : le trafic est décalé par incréments en fonction d'une configuration canarienne, linéaire ou all-at-once de déploiement.
- Déploiements bleu/vert via CloudFormation : le trafic est transféré de vos ressources actuelles vers vos ressources mises à jour dans le cadre d'une CloudFormation mise à jour de la pile. Actuellement, seuls les blue/green déploiements ECS sont pris en charge.

Pour plus d'informations sur les blue/green déploiements, consultez [Vue d'ensemble d'un blue/green déploiement](#).

Pour plus d'informations sur le déploiement automatique depuis Amazon S3, consultez [Déploiement automatique depuis Amazon S3 à l'aide](#) de CodeDeploy.

## Rubriques

- [Créez un déploiement avec CodeDeploy](#)
- [Afficher les détails CodeDeploy du déploiement](#)
- [Afficher les données du journal pour les déploiements CodeDeploy EC2/sur site](#)
- [Arrêtez un déploiement avec CodeDeploy](#)
- [Redéployez et annulez un déploiement avec CodeDeploy](#)
- [Déployer une application dans un autre AWS compte](#)
- [Utiliser l' CodeDeploy agent pour valider un package de déploiement sur une machine locale](#)

## Créez un déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, les API ou les CodeDeploy API pour créer un déploiement qui installe les révisions d'applications que vous avez déjà transférées vers Amazon S3 ou, si votre déploiement concerne une plateforme de calcul EC2/sur site GitHub, sur les instances d'un groupe de déploiement.

Le processus de création d'un déploiement dépend de la plate-forme de calcul utilisée par votre déploiement.

## Rubriques

- [Conditions préalables au déploiement](#)
- [Création d'un déploiement d'Amazon ECS Compute Platform \(console\)](#)
- [Création d'un déploiement de plateforme de calcul AWS Lambda \(console\)](#)
- [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#)
- [Création d'un déploiement de plate-forme de calcul Amazon ECS \(CLI\)](#)
- [Création d'un déploiement de plateforme de calcul AWS Lambda \(CLI\)](#)
- [Création d'un déploiement de plate-forme de calcul EC2/sur site \(CLI\)](#)
- [Créez un blue/green déploiement Amazon ECS via CloudFormation](#)

## Conditions préalables au déploiement

Assurez-vous d'effectuer les étapes suivantes avant de lancer un déploiement.

### Conditions préalables au déploiement sur une plateforme AWS Lambda informatique

- Créez une application qui inclut au moins un groupe de déploiement. Pour plus d'informations, consultez [Créez une application avec CodeDeploy](#) et [Créez un groupe de déploiement avec CodeDeploy](#).
- Préparez la révision de l'application, également appelée AppSpec fichier, qui spécifie la version de la fonction Lambda que vous souhaitez déployer. Le AppSpec fichier peut également spécifier des fonctions Lambda pour valider votre déploiement. Pour de plus amples informations, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).
- Si vous souhaitez utiliser une configuration de déploiement personnalisée pour votre déploiement, créez-la avant de commencer le processus de déploiement. Pour plus d'informations, consultez [Create a Deployment Configuration](#).

### Conditions préalables au déploiement sur une plate-forme informatique EC2/sur site

- Pour un déploiement sur place, créez ou configurez les instances sur lesquelles vous souhaitez effectuer le déploiement. Pour plus d'informations, consultez [Utilisation d'instances pour](#)

[CodeDeploy](#). Pour un blue/green déploiement, vous avez soit un groupe Amazon EC2 Auto Scaling existant à utiliser comme modèle pour votre environnement de remplacement, soit une ou plusieurs instances ou groupes Amazon EC2 Auto Scaling que vous spécifiez comme environnement d'origine. Pour plus d'informations, consultez [Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling](#) et [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).

- Créez une application qui inclut au moins un groupe de déploiement. Pour plus d'informations, consultez [Créez une application avec CodeDeploy](#) et [Créez un groupe de déploiement avec CodeDeploy](#).
- Préparez la révision d'application que vous souhaitez déployer vers les instances de votre groupe de déploiement. Pour plus d'informations, consultez [Travailler avec les révisions d'applications pour CodeDeploy](#).
- Si vous souhaitez utiliser une configuration de déploiement personnalisée pour votre déploiement, créez-la avant de commencer le processus de déploiement. Pour plus d'informations, consultez [Create a Deployment Configuration](#).
- Si vous déployez la révision de votre application à partir d'un compartiment Amazon S3, celui-ci se trouve dans la même AWS région que les instances de votre groupe de déploiement.
- Si vous déployez la révision de votre application à partir d'un compartiment Amazon S3, une politique de compartiment Amazon S3 a été appliquée au compartiment. Cette stratégie accorde à vos instances les autorisations requises pour télécharger la révision d'application.

Par exemple, la politique de compartiment Amazon S3 suivante permet à toute instance Amazon EC2 associée à un profil d'instance IAM contenant l'ARN de `arn:aws:iam::444455556666:role/CodeDeployDemo` télécharger n'importe où dans le compartiment Amazon S3 nommé : `amzn-s3-demo-bucket`

```
{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:role/CodeDeployDemo"
]
 }
 }
]
}
```

```

 }
 }
]
}

```

La politique de compartiment Amazon S3 suivante permet à toute instance sur site associée à un utilisateur IAM contenant l'ARN de `arn:aws:iam::444455556666:user/CodeDeployUser` télécharger depuis n'importe quel emplacement du compartiment Amazon S3 nommé : `amzn-s3-demo-bucket`

```

{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:user/CodeDeployUser"
]
 }
 }
]
}

```

Pour plus d'informations sur la façon de générer et d'associer une politique de compartiment Amazon S3, consultez les [exemples de politique de compartiment](#).

- Si vous créez un blue/green déploiement, ou si vous avez spécifié un Classic Load Balancer, une Application Load Balancer ou un Network Load Balancer en option dans le groupe de déploiement pour un déploiement sur place, vous avez créé un VPC à l'aide d'Amazon VPC contenant au moins deux sous-réseaux. (CodeDeploy utilise Elastic Load Balancing, qui exige que toutes les instances d'un groupe d'équilibreurs de charge se trouvent dans un seul VPC.)

Si vous n'avez pas encore créé de VPC, consultez le guide de [démarrage Amazon VPC](#).

- Si vous créez un blue/green déploiement, vous avez configuré au moins un Classic Load Balancer, un Application Load Balancer ou un Network Load Balancer dans Elastic Load Balancing et vous l'avez utilisé pour enregistrer les instances qui constituent votre environnement d'origine.

 Note

Les instances de votre environnement de remplacement sont inscrites auprès de l'équilibreur de charge plus tard.

Pour plus d'informations sur la configuration d'un équilibreur de charge [Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2](#), reportez-vous aux sections et [Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS](#).

## Conditions préalables au déploiement pour un blue/green déploiement via CloudFormation

- Votre modèle n'a pas besoin de modéliser les ressources d'une CodeDeploy application ou d'un groupe de déploiement.
- Votre modèle doit inclure des ressources pour un VPC utilisant Amazon VPC contenant au moins deux sous-réseaux.
- Votre modèle doit inclure des ressources pour un ou plusieurs équilibreurs de charge classiques, applicatifs ou équilibreurs de charge réseau dans Elastic Load Balancing utilisés pour diriger le trafic vers vos groupes cibles.

## Création d'un déploiement d'Amazon ECS Compute Platform (console)

Cette rubrique explique comment déployer un service Amazon ECS à l'aide de la console. Pour plus d'informations, consultez [Tutoriel : Déployer une application dans Amazon ECS](#) et [Tutoriel : Déployer un service Amazon ECS avec un test de validation](#).

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

## 2. Effectuez l'une des actions suivantes :

- Si vous souhaitez déployer une application, dans le volet de navigation, développez Déployer, puis choisissez Applications. Choisissez le nom de l'application que vous souhaitez déployer. Assurez-vous que la colonne Plateforme de calcul de votre application est Amazon ECS.
- Si vous souhaitez redéployer un déploiement dans le volet de navigation, développez Déployer, puis choisissez Déploiements. Choisissez le déploiement que vous souhaitez redéployer et, dans la colonne Application, choisissez le nom de son application. Assurez-vous que la colonne Plateforme de calcul de votre déploiement est Amazon ECS.

## 3. Sur l'onglet Déploiements, choisissez Créer un déploiement.

### Note

Votre application doit disposer d'un groupe de déploiement avant de pouvoir être déployée. Si votre application ne possède pas de groupe de déploiement, dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement. Pour de plus amples informations, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#).

## 4. Dans Groupe de déploiement, choisissez un groupe de déploiement à utiliser pour ce déploiement.

## 5. En regard de Emplacement de la révision, choisissez l'emplacement où est stockée votre révision :

- Mon application est stockée dans Amazon S3. Pour plus d'informations, reportez-vous à l'étape 6 [Spécifiez les informations relatives à une révision stockée dans un compartiment Amazon S3](#), puis revenez à cette étape.
- Utiliser l' AppSpec éditeur : sélectionnez JSON ou YAML, puis entrez votre AppSpec fichier dans l'éditeur. Vous pouvez enregistrer le AppSpec fichier en choisissant Enregistrer sous forme de fichier texte. Lorsque vous choisissez Déployer à la fin de ces étapes, vous recevez une erreur si votre code JSON ou YAML n'est pas valide. Pour plus d'informations sur la création d'un AppSpec fichier, consultez [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).

## 6. (Facultatif) Dans la zone Description du déploiement, entrez une description pour ce déploiement.

7. (Facultatif) Dans Remplacements de la configuration de restauration, vous pouvez spécifier pour ce déploiement des options de restauration automatique différentes de celles spécifiées pour le groupe de déploiement, le cas échéant.

Pour plus d'informations sur les annulations CodeDeploy, reportez-vous aux sections [Redéploiements et annulations de déploiements](#) et [Redéployez et annulez un déploiement avec CodeDeploy](#).

Choisissez parmi les options suivantes :

- Annulation en cas d'échec d'un déploiement : CodeDeploy redéploie la dernière révision valide connue en tant que nouveau déploiement.
  - Annulation lorsque les seuils d'alarme sont atteints : si des alarmes ont été ajoutées au groupe de déploiement, CodeDeploy redéploie la dernière révision valide connue lorsqu'une ou plusieurs des alarmes spécifiées sont activées.
  - Désactiver les annulations : n'effectuez pas de annulations pour ce déploiement.
8. Choisissez Créer un déploiement.

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## Création d'un déploiement de plateforme de calcul AWS Lambda (console)

Cette rubrique explique comment déployer une fonction Lambda à l'aide de la console.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Effectuez l'une des actions suivantes :
  - Si vous souhaitez déployer une application, dans le volet de navigation, développez Déployer, puis choisissez Applications. Choisissez le nom de l'application que vous souhaitez déployer. Assurez-vous que la colonne Plateforme de calcul de votre application est AWS Lambda.



- Si vous souhaitez redéployer un déploiement dans le volet de navigation, développez Déployer, puis choisissez Déploiements. Choisissez le déploiement que vous souhaitez redéployer et, dans la colonne Application, choisissez le nom de son application. Assurez-vous que la colonne Plateforme de calcul de votre déploiement est AWS Lambda.
3. Sur l'onglet Déploiements, choisissez Créer un déploiement.

 Note

Votre application doit disposer d'un groupe de déploiement avant de pouvoir être déployée. Si votre application ne possède pas de groupe de déploiement, dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement. Pour de plus amples informations, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#).

4. Dans Groupe de déploiement, choisissez un groupe de déploiement à utiliser pour ce déploiement.
5. En regard de Emplacement de la révision, choisissez l'emplacement où est stockée votre révision :
  - Mon application est stockée dans Amazon S3. Pour plus d'informations, reportez-vous à l'étape 6 [Spécifiez les informations relatives à une révision stockée dans un compartiment Amazon S3](#), puis revenez à cette étape.
  - Utiliser l' AppSpec éditeur : sélectionnez JSON ou YAML, puis entrez votre AppSpec fichier dans l'éditeur. Vous pouvez enregistrer le AppSpec fichier en choisissant Enregistrer sous forme de fichier texte. Lorsque vous choisissez Déployer à la fin de ces étapes, vous recevez une erreur si votre code JSON ou YAML n'est pas valide. Pour plus d'informations sur la création d'un AppSpec fichier, consultez [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#).
6. (Facultatif) Dans la zone Description du déploiement, entrez une description pour ce déploiement.
7. (Facultatif) Développez les remplacements des groupes de déploiement pour choisir une configuration de déploiement afin de contrôler la manière dont le trafic est transféré vers la version de la fonction Lambda différente de celle spécifiée dans le groupe de déploiement.

Pour de plus amples informations, veuillez consulter [Configurations de déploiement sur une plateforme AWS Lambda informatique](#).

8. (Facultatif) Dans Remplacements de la configuration de restauration, vous pouvez spécifier pour ce déploiement des options de restauration automatique différentes de celles spécifiées pour le groupe de déploiement, le cas échéant.

Pour plus d'informations sur les annulations CodeDeploy, reportez-vous aux sections [Redéploiements et annulations de déploiements](#) et [Redéployez et annulez un déploiement avec CodeDeploy](#).

Choisissez parmi les options suivantes :

- Annulation en cas d'échec d'un déploiement : CodeDeploy redéploie la dernière révision valide connue en tant que nouveau déploiement.
  - Annulation lorsque les seuils d'alarme sont atteints : si des alarmes ont été ajoutées au groupe de déploiement, CodeDeploy redéploie la dernière révision valide connue lorsqu'une ou plusieurs des alarmes spécifiées sont activées.
  - Désactiver les annulations : n'effectuez pas de annulations pour ce déploiement.
9. Choisissez Créer un déploiement.

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## Création d'un déploiement de plate-forme de calcul EC2/sur site (console)

Cette rubrique explique comment déployer une application sur un serveur Amazon EC2 ou sur site à l'aide de la console.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Effectuez l'une des actions suivantes :

- Si vous souhaitez déployer une application, dans le volet de navigation, développez Déployer, puis choisissez Applications. Choisissez le nom de l'application que vous souhaitez déployer. Assurez-vous que la colonne Plateforme de calcul de votre application est EC2/On-premises.
  - Si vous souhaitez redéployer un déploiement dans le volet de navigation, développez Déployer, puis choisissez Déploiements. Recherchez le déploiement que vous souhaitez redéployer, puis choisissez le nom de son application dans la colonne Application. Assurez-vous que la colonne Plateforme de calcul de votre déploiement est EC2/on-premises.
3. Sur l'onglet Déploiements, choisissez Créer un déploiement.

 Note

Votre application doit disposer d'un groupe de déploiement avant de pouvoir être déployée. Si votre application ne possède pas de groupe de déploiement, dans l'onglet Groupes de déploiement, choisissez Créer un groupe de déploiement. Pour de plus amples informations, veuillez consulter [Créez un groupe de déploiement avec CodeDeploy](#).

4. Dans Groupe de déploiement, choisissez un groupe de déploiement à utiliser pour ce déploiement.
5. En regard de Type de référentiel, choisissez le type de référentiel dans lequel la révision est stockée :
- Mon application est stockée dans Amazon S3. Pour plus d'informations, reportez-vous à l'étape 6 [Spécifiez les informations relatives à une révision stockée dans un compartiment Amazon S3](#), puis revenez à cette étape.
  - Ma demande est stockée dans GitHub — Pour plus d'informations, reportez-vous à l'étape 6 [Spécifier les informations relatives à une révision stockée dans un GitHub référentiel](#), puis revenez à l'étape 6.
6. (Facultatif) Dans la zone Description du déploiement, entrez une description pour ce déploiement.
7. (Facultatif) Développez la configuration de déploiement Override pour choisir une configuration de déploiement différente de celle spécifiée dans le groupe de déploiement afin de contrôler la manière dont le trafic est transféré vers le serveur Amazon EC2 ou sur site.

Pour de plus amples informations, veuillez consulter [Utilisation des configurations de déploiement dans CodeDeploy](#).

8.
  - a. Sélectionnez Ne pas échouer le déploiement si l'événement du ApplicationStop cycle de vie échoue si vous souhaitez que le déploiement sur une instance réussisse en cas d'échec de l'événement ApplicationStop du cycle de vie.
  - b. Développez les paramètres de comportement de déploiement supplémentaires pour spécifier CodeDeploy le mode de gestion des fichiers dans un emplacement cible de déploiement qui ne faisaient pas partie du déploiement réussi précédent.

Choisissez parmi les options suivantes :

- Échec du déploiement : une erreur est signalée et l'état du déploiement est changé enFailed.
- Remplacer le contenu : si un fichier du même nom existe à l'emplacement cible, la version de la révision de l'application le remplace.
- Conserver le contenu : si un fichier du même nom existe à l'emplacement cible, le fichier est conservé et la version de la révision de l'application n'est pas copiée sur l'instance.

Pour de plus amples informations, veuillez consulter [Comportement d'annulation avec du contenu existant](#).

9. (Facultatif) Dans Remplacements de la configuration de restauration, vous pouvez spécifier pour ce déploiement des options de restauration automatique différentes de celles spécifiées pour le groupe de déploiement, le cas échéant.

Pour plus d'informations sur les annulations CodeDeploy, reportez-vous aux sections [Redéploiements et annulations de déploiements](#) et [Redéployez et annulez un déploiement avec CodeDeploy](#).

Choisissez parmi les options suivantes :

- Annulation en cas d'échec d'un déploiement : CodeDeploy redéploie la dernière révision valide connue en tant que nouveau déploiement.
- Annulation lorsque les seuils d'alarme sont atteints : si des alarmes ont été ajoutées au groupe de déploiement, CodeDeploy déploie la dernière révision correcte connue lorsqu'une ou plusieurs des alarmes spécifiées sont activées.
- Désactiver les annulations : n'effectuez pas de annulations pour ce déploiement.

10. Choisissez Démarrer le déploiement.

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## Rubriques

- [Spécifiez les informations relatives à une révision stockée dans un compartiment Amazon S3](#)
- [Spécifier les informations relatives à une révision stockée dans un GitHub référentiel](#)

## Spécifiez les informations relatives à une révision stockée dans un compartiment Amazon S3

Si vous suivez les étapes décrites dans [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#), suivez ces étapes pour ajouter des informations sur une révision d'application stockée dans un compartiment Amazon S3.

1. Copiez le lien Amazon S3 de votre révision dans l'emplacement de la révision. Pour trouver la valeur du lien :
  - a. Dans un onglet de navigateur distinct :

Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/s3/>.

Recherchez et choisissez votre révision.
  - b. Si le volet Propriétés n'est pas visible, choisissez le bouton Propriétés.
  - c. Dans le volet Propriétés, copiez la valeur du champ Lien dans le champ Emplacement des révisions de la CodeDeploy console.

Pour spécifier un ETag (une somme de contrôle de fichier) dans le cadre de l'emplacement des révisions :

- Si la valeur du champ Lien se termine par **?versionId=versionId**, ajoutez **&etag=** et ETag à la fin de la valeur du champ Lien.
- Si la valeur du champ Lien ne spécifie pas d'ID de version, ajoutez **?etag=** et ETag à la fin de la valeur du champ Lien.

 Note

Cela n'est pas aussi aisé que de copier la valeur du champ Lien, mais vous pouvez également entrer l'emplacement de la révision dans l'un des formats suivants :

**s3://*bucket-name*/*folders*/*objectName***

**s3://*bucket-name*/*folders*/*objectName*?versionId=*versionId***

**s3://*bucket-name*/*folders*/*objectName*?etag=*etag***

**s3://*bucket-name*/*folders*/*objectName*?versionId=*versionId*&etag=*etag***

***bucket-name*.s3.amazonaws.com/*folders*/*objectName***

2. Si un message s'affiche dans la liste Type de fichier, indiquant que le type de fichier n'a pas pu être détecté, choisissez le type de fichier de la révision. Sinon, acceptez le type de fichier détecté.

## Spécifier les informations relatives à une révision stockée dans un GitHub référentiel

Si vous suivez les étapes décrites dans [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#), procédez comme suit pour ajouter des informations sur une révision d'application stockée dans un GitHub référentiel.

1. Dans Connect to GitHub, effectuez l'une des opérations suivantes :
  - Pour créer une connexion entre CodeDeploy les applications et un GitHub compte, dans un autre onglet du navigateur Web, déconnectez-vous GitHub. Dans le GitHub compte, entrez un nom pour identifier cette connexion, puis choisissez Connect to GitHub. La page Web vous invite CodeDeploy à autoriser l'interaction avec GitHub votre application. Passez à l'étape 2.
  - Pour utiliser une connexion que vous avez déjà créée, dans le GitHub compte, sélectionnez son nom, puis choisissez Connect to GitHub. Passez à l'étape 4.
  - Pour créer une connexion à un autre GitHub compte, dans un autre onglet du navigateur Web, déconnectez-vous GitHub. Choisissez Se connecter à un autre GitHub compte, puis sélectionnez Connect to GitHub. Passez à l'étape 2.
2. Si vous êtes invité à vous connecter à GitHub, suivez les instructions de la page de connexion. Connectez-vous à l'aide de votre nom GitHub d'utilisateur ou de votre adresse e-mail et de votre mot de passe.
3. Si une page Authorize application s'affiche, choisissez Authorize application.

4. Sur la page Créer un déploiement, dans le champ Nom du référentiel, entrez le nom de l' GitHub utilisateur ou de l'organisation contenant la révision, suivi d'une barre oblique (/), puis du nom du référentiel contenant la révision. Si vous n'êtes pas sûr de la valeur à saisir :
  - a. Dans un autre onglet de navigateur Web, accédez à votre [GitHub tableau de bord](#).
  - b. Dans Your repositories, placez le pointeur de la souris sur le nom du référentiel cible. Une infobulle apparaît, affichant le nom de l' GitHub utilisateur ou de l'organisation, suivi d'une barre oblique (/), puis du nom du référentiel. Saisissez la valeur affichée dans la zone Nom du référentiel.

 Note

Si le nom du référentiel cible n'est pas visible dans Vos référentiels, utilisez la GitHub zone de recherche pour trouver le nom du référentiel cible et le nom de GitHub l'utilisateur ou de l'organisation.

5. Dans la zone ID de validation, saisissez l'ID de la validation qui fait référence à la révision figurant dans le référentiel. Si vous n'êtes pas sûr de la valeur à saisir :
  - a. Dans un autre onglet de navigateur Web, accédez à votre [GitHub tableau de bord](#).
  - b. Dans Your repositories, choisissez le nom de référentiel qui contient la validation cible.
  - c. Dans la liste des validations, recherchez et copiez l'ID de validation qui fait référence à la révision figurant dans le référentiel. Cet ID contient généralement 40 caractères et se compose de lettres et de chiffres. (N'utilisez pas la version courte de l'ID de validation, qui correspond généralement aux 10 premiers caractères de la version longue.)
  - d. Collez l'ID de validation dans la zone ID de validation.

## Création d'un déploiement de plate-forme de calcul Amazon ECS (CLI)

Après avoir créé l'application et la révision (dans les déploiements Amazon ECS, il s'agit du AppSpec fichier) :

Appelez la commande [create-deployment](#) en spécifiant :

- Un nom d'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).

- Un nom de groupe de déploiement. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).
- Des informations sur la révision à déployer :

Pour les révisions stockées dans Amazon S3 :

- Le nom du compartiment Amazon S3 qui contient la révision.
- Le nom de la révision chargée.
- (Facultatif) L'identifiant de version Amazon S3 pour la révision. (Si l'identifiant de version n'est pas spécifié, CodeDeploy utilise la version la plus récente.)
- (Facultatif) Le ETag pour la révision. (Si le n' ETag est pas spécifié, CodeDeploy ignore la validation de l'objet.)

Pour les révisions stockées dans un fichier qui ne figure pas dans Amazon S3, vous avez besoin du nom du fichier et de son chemin. Votre fichier de révision est écrit en JSON ou YAML, il est donc très probable que son extension soit .json ou .yaml.

- (Facultatif) Une description du déploiement.

Le fichier de révision peut être spécifié sous forme de fichier chargé dans un compartiment Amazon S3 ou sous forme de chaîne. La syntaxe des deux options dans le cadre d'une utilisation de la commande `create-deployment` est :

- Compartiment Amazon S3 :

La `version` et l'`eTag` sont facultatifs.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Chaîne :

```
--revision '{"revisionType": "String", "string": {"content": "revision-as-string"}}'
```



 Note

La commande `create-deployment` permet de charger une révision à partir d'un fichier. Pour de plus amples informations, veuillez consulter [Chargement des paramètres à partir d'un fichier](#).

Pour les modèles de révision de AWS Lambda déploiement, voir [Ajouter un AppSpec fichier pour un déploiement AWS Lambda](#). Pour obtenir un exemple de révision, consultez [AppSpec Exemple de fichier pour un déploiement AWS Lambda](#).

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## Création d'un déploiement de plateforme de calcul AWS Lambda (CLI)

Après avoir créé l'application et la révision (dans les déploiements AWS Lambda, il s'agit du AppSpec fichier) :

Appelez la commande [create-deployment](#) en spécifiant :

- Un nom d'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Un nom de groupe de déploiement. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).
- Des informations sur la révision à déployer :

Pour les révisions stockées dans Amazon S3 :

- Le nom du compartiment Amazon S3 qui contient la révision.
- Le nom de la révision chargée.
- (Facultatif) L'identifiant de version Amazon S3 pour la révision. (Si l'identifiant de version n'est pas spécifié, CodeDeploy utilise la version la plus récente.)
- (Facultatif) Le ETag pour la révision. (Si le n' ETag est pas spécifié, CodeDeploy ignore la validation de l'objet.)

Pour les révisions stockées dans un fichier qui ne figure pas dans Amazon S3, vous avez besoin du nom du fichier et de son chemin. Votre fichier de révision est écrit en JSON ou YAML, il est donc très probable que son extension soit `.json` ou `.yaml`.

- (Facultatif) Le nom d'une configuration de déploiement à utiliser. Pour afficher la liste des configurations de déploiement, appelez la commande [list-deployment-configs](#). (Si ce n'est pas spécifié, CodeDeploy utilise une configuration de déploiement par défaut spécifique.)
- (Facultatif) Une description du déploiement.

Le fichier de révision peut être spécifié sous forme de fichier chargé dans un compartiment Amazon S3 ou sous forme de chaîne. La syntaxe des deux options dans le cadre d'une utilisation de la commande `create-deployment` est :

- Compartiment Amazon S3 :

La `version` et l'`eTag` sont facultatifs.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Chaîne :

```
--revision '{"revisionType": "String", "string": {"content": "revision-as-string"}}'
```

#### Note

La commande `create-deployment` permet de charger une révision à partir d'un fichier. Pour de plus amples informations, veuillez consulter [Chargement des paramètres à partir d'un fichier](#).

Pour les modèles de révision de AWS Lambda déploiement, voir [Ajouter un AppSpec fichier pour un déploiement AWS Lambda](#). Pour obtenir un exemple de révision, consultez [AppSpec Exemple de fichier pour un déploiement AWS Lambda](#).

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## Création d'un déploiement de plate-forme de calcul EC2/sur site (CLI)

Pour utiliser le AWS CLI pour déployer une révision de la plate-forme de calcul EC2/sur site :

1. Une fois que vous avez préparé les instances, créé l'application et transmis la révision, effectuez l'une des actions suivantes :
  - Si vous souhaitez déployer une révision à partir d'un compartiment Amazon S3, passez à l'étape 2 dès maintenant.
  - Si vous souhaitez déployer une révision à partir d'un GitHub référentiel, suivez d'abord les étapes décrites [Connect une CodeDeploy application à un GitHub référentiel](#), puis passez à l'étape 2.
2. Appelez la commande [create-deployment](#) en spécifiant :
  - `--application-name`: nom de l'application. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
  - `--deployment-group-name`: nom du groupe de déploiement Amazon EC2. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).
  - `--revision`: Informations sur la révision à déployer :

Pour les révisions stockées dans Amazon S3 :

- `s3Location`: le nom du compartiment Amazon S3 qui contient la révision.
- `s3Location --> key`: nom de la révision téléchargée.
- `s3Location --> bundleType`: le type de fichier de la révision téléchargée.

 Note

Les formats de fichier d'archive tar et tar compressé (.tar et .tar.gz) ne sont pas pris en charge pour les instances Windows Server.

- `s3Location --> version`: (Facultatif) L'identifiant de version Amazon S3 pour la révision. (Si l'identifiant de version n'est pas spécifié, CodeDeploy utilise la version la plus récente.)
- `s3Location --> eTag`: (Facultatif) Le ETag pour la révision. (Si le n' ETag est pas spécifié, CodeDeploy ignore la validation de l'objet.)

Pour les révisions stockées dans GitHub :

- `githubLocation --> repository`: nom GitHub d'utilisateur ou de groupe attribué au référentiel contenant la révision, suivi d'une barre oblique (/), puis du nom du référentiel.
- `githubLocation --> commitId`: ID de validation pour la révision.
- `--deployment-config-name`: (Facultatif) Le nom de la configuration de déploiement

[list-deployment-configs](#). (Si ce n'est pas spécifié, CodeDeploy utilise une configuration de déploiement par défaut spécifique.)

- `--ignore-application-stop-failures | --no-ignore-application-stop-failures`: (Facultatif) Si vous souhaitez que le déploiement sur une instance se poursuive jusqu'à l'événement du cycle de vie du `BeforeInstall` déploiement en cas d'échec de l'événement du cycle de vie du `ApplicationStop` déploiement.
- `--description`: (Facultatif) Description du déploiement.
- `--file-exists-behavior`: (Facultatif) Dans le cadre du processus de déploiement, l'CodeDeploy agent supprime de chaque instance tous les fichiers installés lors du déploiement le plus récent. Choisissez ce qui se passe lorsque des fichiers qui ne faisaient pas partie d'un déploiement précédent apparaissent dans les emplacements de déploiement cibles.
- `--target-instances`: pour blue/green les déploiements, informations sur les instances appartenant à l'environnement de remplacement dans un blue/green déploiement, y compris les noms d'un ou de plusieurs groupes Amazon EC2 Auto Scaling, ou la clé de filtre de balise, le type et la valeur utilisés pour identifier les instances Amazon EC2.

#### Note

Utilisez cette syntaxe dans le cadre de l'`create-deployment` appel pour spécifier les informations relatives à une révision dans Amazon S3 directement sur la ligne de commande. (`version` et `eTag` sont facultatifs.)

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|
zip,version=string,eTag=string
```

Utilisez cette syntaxe dans le cadre de l'`create-deployment` appel pour spécifier les informations relatives à une révision GitHub directement sur la ligne de commande :

```
--github-location repository=string,commitId=string
```

Pour obtenir des informations sur les révisions qui ont déjà été transmises en mode push, appelez la commande [list-application-revisions](#).

Pour suivre le statut de votre déploiement, consultez [Afficher les détails CodeDeploy du déploiement](#).

## référence de commande create-deployment

Vous trouverez ci-dessous la structure de commande et les options de la `create-deployment` commande. Pour plus d'informations, consultez la référence de [création et de déploiement](#) dans la référence des AWS CLI commandes.

```
create-deployment
--application-name <value>
[--deployment-group-name <value>]
[--revision <value>]
[--deployment-config-name <value>]
[--description <value>]
[--ignore-application-stop-failures | --no-ignore-application-stop-failures]
[--target-instances <value>]
[--auto-rollback-configuration <value>]
[--update-outdated-instances-only | --no-update-outdated-instances-only]
[--file-exists-behavior <value>]
[--s3-location <value>]
[--github-location <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

## Connect une CodeDeploy application à un GitHub référentiel

Avant de déployer une application à partir d'un GitHub référentiel pour la première fois à l'aide du AWS CLI, vous devez d'abord CodeDeploy autoriser l'interaction GitHub au nom de votre GitHub compte. Cette étape doit être effectuée une fois pour chaque application utilisant la CodeDeploy console.

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Choisissez Applications.

3. Dans Applications, choisissez l'application que vous souhaitez associer à votre compte GitHub utilisateur, puis choisissez Déployer l'application.

 Note

Vous ne créez pas de déploiement. C'est actuellement le seul moyen de donner CodeDeploy l'autorisation d'interagir avec vous GitHub au nom de votre compte GitHub utilisateur.

4. À côté de Type de référentiel, choisissez Ma révision d'application est stockée dans GitHub.
5. Choisissez Connect to GitHub.

 Note

Si le lien Connect to a different account (Connexion à un autre GitHub compte) s'affiche : Vous êtes peut-être déjà autorisé CodeDeploy à interagir avec GitHub l'application au nom d'un autre GitHub compte. Vous avez peut-être révoqué l'autorisation CodeDeploy d'interagir avec GitHub le compte connecté pour toutes les applications liées à ce GitHub compte. CodeDeploy Pour de plus amples informations, veuillez consulter [GitHub authentication avec des applications dans CodeDeploy](#).

6. Si vous n'êtes pas encore connecté à GitHub, suivez les instructions de la page de connexion.
7. Sur la page Authorize application, choisissez Authorize application.
8. Maintenant CodeDeploy que vous y êtes autorisé, choisissez Annuler et poursuivez les étapes ci-dessous [Création d'un déploiement de plate-forme de calcul EC2/sur site \(CLI\)](#).

## Créez un blue/green déploiement Amazon ECS via CloudFormation

Vous pouvez l'utiliser AWS CloudFormation pour gérer les blue/green déploiements Amazon ECS via CodeDeploy. Vous générez votre déploiement en définissant vos ressources vertes et bleues et en spécifiant les paramètres de routage et de stabilisation du trafic à utiliser dans CloudFormation. Cette rubrique décrit les différences entre les blue/green déploiements Amazon ECS gérés par CodeDeploy et les déploiements gérés par CloudFormation

Pour step-by-step obtenir des instructions sur la manière CloudFormation de gérer vos déploiements bleu/vert Amazon ECS, consultez la section [Automatiser les déploiements bleu/vert d'ECS dans CodeDeploy le guide de l'utilisateur](#). AWS CloudFormationAWS CloudFormation

 Note

La gestion des blue/green déploiements Amazon ECS avec n' CloudFormation est pas disponible dans la région Asie-Pacifique (Osaka).

## Différences entre les blue/green déploiements Amazon ECS via CodeDeploy et CloudFormation

Le modèle de CloudFormation pile modélise les ressources et l'infrastructure liées aux tâches Amazon ECS, ainsi que les options de configuration pour les déploiements. Il existe donc des différences entre les déploiements Amazon ECS standard et blue/green les blue/green déploiements créés via. CloudFormation

Contrairement aux blue/green déploiements Amazon ECS standard, vous ne modélisez ni ne créez manuellement les éléments suivants :

- Vous ne créez pas une AWS CodeDeploy application en spécifiant un nom qui représente de manière unique ce que vous souhaitez déployer.
- Vous ne créez pas de groupe AWS CodeDeploy de déploiement.
- Vous ne spécifiez pas de fichier de spécification d'application (AppSpec fichier). Les informations normalement gérées avec le AppSpec fichier, telles que les options de configuration pondérées ou les événements du cycle de vie, sont gérées par le `AWS::CodeDeploy::BlueGreen` hook.

Ce tableau récapitule les différences dans le flux de travail de haut niveau entre les types de déploiement.

| Fonction                                                                          | blue/green Déploiements standard                                       | Déploiements bleu/vert via CloudFormation                     |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------|
| Spécifiez le cluster Amazon ECS, le service Amazon ECS, Application Load Balancer | Créez un groupe CodeDeploy de déploiement qui spécifie ces ressources. | Créez un CloudFormation modèle pour modéliser ces ressources. |

| Fonction                                                                                                 | blue/green Déploiements standard                           | Déploiements bleu/vert via CloudFormation                                                                                |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| ou Network Load Balancer, l'écouteur de production, l'écouteur de test et deux groupes cibles.           |                                                            |                                                                                                                          |
| Spécifiez la modification à déployer.                                                                    | Créez une CodeDeploy application.                          | Créez un CloudFormation modèle qui spécifie l'image du conteneur.                                                        |
| Spécifiez la définition de tâche Amazon ECS, le nom du conteneur et le port du conteneur.                | Créez un AppSpec fichier qui spécifie ces ressources.      | Créez un CloudFormation modèle pour modéliser ces ressources.                                                            |
| Spécifiez les options de déplacement du trafic de déploiement et les hooks d'événements de cycle de vie. | Créez un AppSpec fichier qui spécifie ces options.         | Créez un CloudFormation modèle qui utilise les paramètres du AWS::CodeDeploy::BlueGreen hook pour spécifier ces options. |
| CloudWatch alarmes.                                                                                      | Créez une CloudWatch alarme qui déclenche une annulation.  | Configurez une CloudWatch alarme au niveau de la CloudFormation pile qui déclenche une annulation.                       |
| Restauration/redéploiement.                                                                              | Spécifiez les options de restauration et de redéploiement. | Annulez la mise à jour de la pile dans CloudFormation.                                                                   |

## Surveillance des blue/green déploiements Amazon ECS via CloudFormation

Vous pouvez surveiller les blue/green déploiements via CloudFormation et CodeDeploy. Pour plus d'informations sur la surveillance via CloudFormation, consultez la section [Surveillance blue/green des événements CloudFormation dans](#) le guide de AWS CloudFormation l'utilisateur.



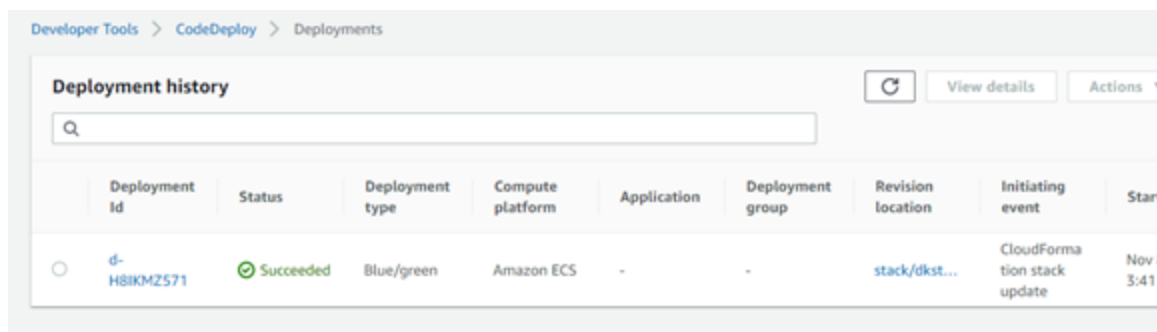
## Pour consulter l'état de déploiement des blue/green déploiements dans CodeDeploy

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

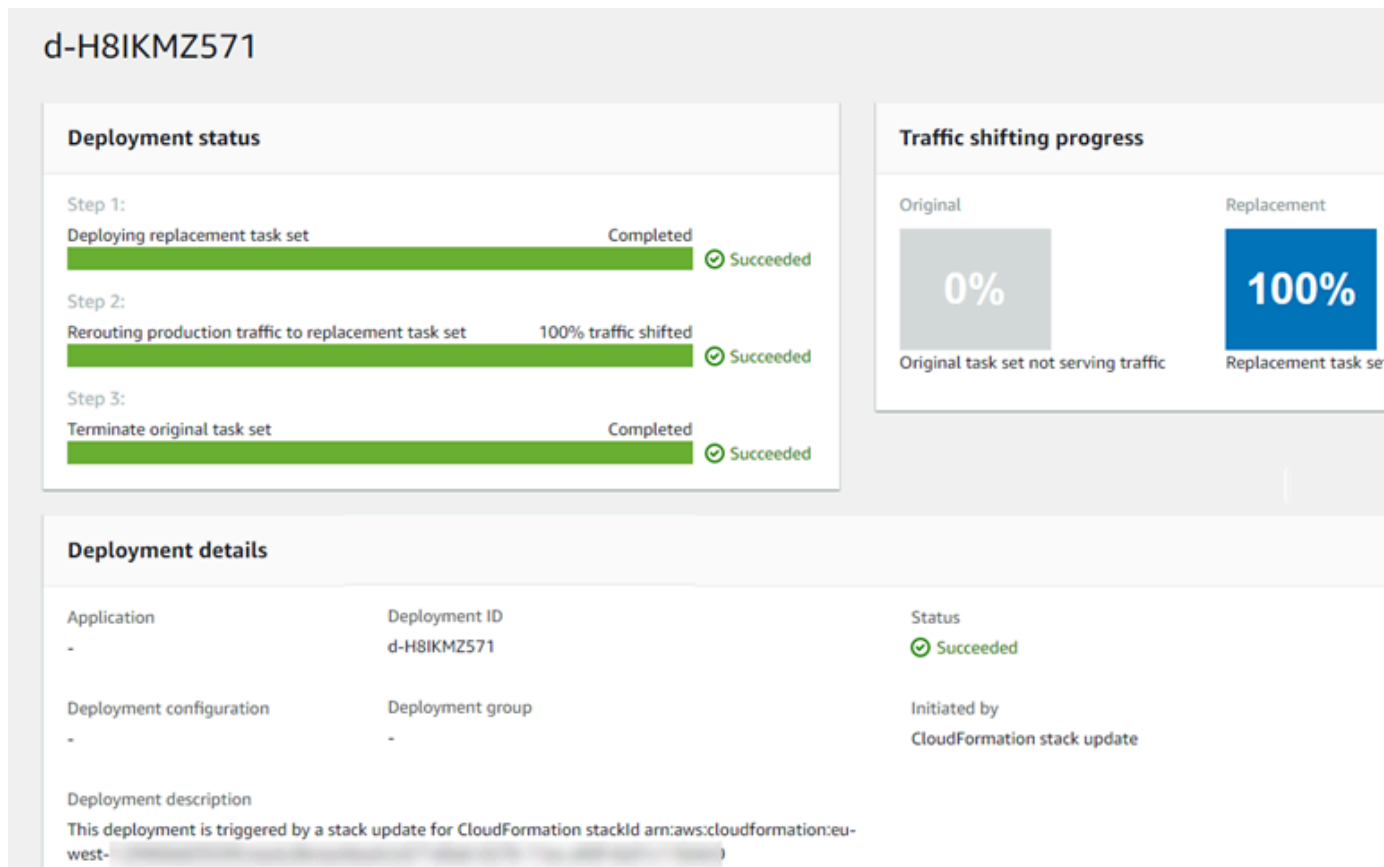
2. Dans Déploiements, le déploiement qui a été déclenché par la mise à jour de la CloudFormation pile apparaît. Choisissez le déploiement pour afficher l'historique des déploiements.



The screenshot shows the AWS CodeDeploy console interface. At the top, there's a breadcrumb trail: "Developer Tools > CodeDeploy > Deployments". Below this is a "Deployment history" section with a search bar and buttons for "Refresh", "View details", and "Actions". The main part of the screenshot is a table with the following columns: Deployment Id, Status, Deployment type, Compute platform, Application, Deployment group, Revision location, Initiating event, and Start. There is one row of data showing a deployment with ID "d-H8IKMZ571", Status "Succeeded", Deployment type "Blue/green", Compute platform "Amazon ECS", and Initiating event "CloudFormation stack update".

| Deployment Id | Status    | Deployment type | Compute platform | Application | Deployment group | Revision location | Initiating event            | Start      |
|---------------|-----------|-----------------|------------------|-------------|------------------|-------------------|-----------------------------|------------|
| d-H8IKMZ571   | Succeeded | Blue/green      | Amazon ECS       | -           | -                | stack/dkst...     | CloudFormation stack update | Nov 1 3:41 |

3. Choisissez le déploiement pour afficher l'état de déplacement du trafic. Notez que l'application et le groupe de déploiement ne sont pas créés.



4. Les éléments suivants s'appliquent à la restauration ou à l'arrêt du déploiement :

- Le déploiement réussi apparaît dans CodeDeploy et indique que le déploiement a été initié par CloudFormation.
- Si vous souhaitez arrêter et annuler le déploiement, vous devez annuler la mise à jour de la pile dans CloudFormation.

## Afficher les détails CodeDeploy du déploiement

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour afficher les détails des déploiements associés à votre AWS compte.

### Note

Vous pouvez consulter les journaux de déploiement EC2/site sur vos instances aux emplacements suivants :

- Amazon Linux, RHEL et Ubuntu Server : `/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`
- Serveur Windows : `C : \ ProgramData \ Amazon \ CodeDeploy <DEPLOYMENT-GROUP-ID>\ logs \ scripts.log`

Pour de plus amples informations, veuillez consulter [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#).

## Rubriques

- [Afficher les détails du déploiement \(console\)](#)
- [Afficher les détails du déploiement \(CLI\)](#)

## Afficher les détails du déploiement (console)

Pour consulter les détails du déploiement à l'aide de la CodeDeploy console, procédez comme suit :

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Deployments.

### Note

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy n'est pris en charge que dans ces régions.

3. Pour afficher plus de détails sur un déploiement unique, dans Historique des déploiements, choisissez l'ID de déploiement ou le bouton en regard de l'ID de déploiement, puis choisissez Afficher.

## Afficher les détails du déploiement (CLI)

Pour utiliser le AWS CLI pour afficher les détails du déploiement, appelez la `get-deployment` commande ou la `batch-get-deployments` commande. Vous pouvez appeler la `list-deployments` commande pour obtenir une liste des déploiements uniques IDs à utiliser comme entrées pour la `get-deployment` commande et la `batch-get-deployments` commande.

Pour afficher les détails d'un seul déploiement, appelez la commande [get-deployment](#), en spécifiant l'identifiant de déploiement unique. Pour obtenir l'ID de déploiement, appelez la commande [list-deployments](#).

Pour afficher des détails sur plusieurs déploiements, appelez la commande [batch-get-deployments](#), en spécifiant plusieurs identificateurs de déploiement uniques. Pour obtenir le déploiement IDs, appelez la commande [list-deployments](#).

Pour afficher la liste des déploiements IDs, appelez la commande [list-deployments](#), en spécifiant :

- Le nom de l'application associée au déploiement. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Le nom du groupe de déploiement associé au déploiement. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).
- Eventuellement, s'il convient d'inclure des détails sur les déploiements en fonction de leur état de déploiement. (Si vous ne spécifiez pas cela, tous les déploiements correspondants sont répertoriés, quel que soit leur état de déploiement.)
- Eventuellement, s'il convient d'inclure des détails sur les déploiements en fonction de leur heure de début ou de leur heure de fin de création de déploiement, ou des deux. (Si vous ne spécifiez pas cela, tous les déploiements correspondants sont répertoriés, quel que soit leur heure de création.)

# Afficher les données du journal pour les déploiements CodeDeploy EC2/sur site

Vous pouvez consulter les données du journal créées par un CodeDeploy déploiement en configurant l' CloudWatch agent Amazon pour qu'il affiche les données agrégées dans la CloudWatch console ou en vous connectant à une instance individuelle pour consulter le fichier journal.

## Note

Les journaux ne sont pas pris en charge pour les AWS déploiements Lambda ou Amazon ECS. Ils peuvent être créés uniquement pour les déploiements EC2/sur site.

## Rubriques

- [Afficher les données du fichier journal dans la CloudWatch console Amazon](#)
- [Afficher les fichiers journaux d'une instance](#)

## Afficher les données du fichier journal dans la CloudWatch console Amazon

Lorsque l' CloudWatch agent Amazon est installé sur une instance, les données de déploiement de tous les déploiements sur cette instance peuvent être consultées dans la CloudWatch console. Pour des raisons de simplicité, nous vous recommandons CloudWatch de surveiller les fichiers journaux de manière centralisée au lieu de les afficher instance par instance. Pour de plus amples informations, veuillez consulter [Envoyer les journaux des CodeDeploy agents à CloudWatch](#).

## Afficher les fichiers journaux d'une instance

Pour afficher les données du journal de déploiement pour une instance individuelle, vous pouvez vous connecter à l'instance et chercher des informations sur les erreurs ou d'autres événements de déploiement.

## Rubriques

- [Pour consulter les fichiers journaux de déploiement sur les instances Amazon Linux, RHEL et Ubuntu Server](#)
- [Pour afficher les fichiers journaux de déploiement sur les instances Windows Server](#)

## Pour consulter les fichiers journaux de déploiement sur les instances Amazon Linux, RHEL et Ubuntu Server

Sur les instances Amazon Linux, RHEL et Ubuntu Server, les journaux de déploiement sont stockés à l'emplacement suivant :

```
/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
```

Pour consulter ou analyser les journaux de déploiement sur les instances Amazon Linux, RHEL et Ubuntu Server, connectez-vous à l'instance, puis tapez la commande suivante pour ouvrir le fichier journal de l' CodeDeploy agent :

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Tapez les commandes suivantes pour parcourir le fichier journal à la recherche des messages d'erreur :

| Commande           | Résultat                                                                                                                         |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>&amp; ERROR</b> | Affiche uniquement les messages d'erreur dans le fichier journal. Utilisez un espace simple avant et après le mot <b>ERROR</b> . |
| <b>/ ERROR</b>     | Recherchez le message d'erreur suivant. <sup>1</sup>                                                                             |
| <b>? ERROR</b>     | Recherchez le message d'erreur précédent. <sup>2</sup> Utilisez un seul espace avant et après le mot. <b>ERROR</b>               |
| <b>G</b>           | Accède à la fin du fichier journal.                                                                                              |
| <b>g</b>           | Accède au début du fichier journal.                                                                                              |
| <b>q</b>           | Quitte le fichier journal.                                                                                                       |
| <b>h</b>           | Donne des informations sur les commandes supplémentaires.                                                                        |

| Commande                                                                                                                                                   | Résultat |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <sup>1</sup> Après avoir saisi <b>/ ERROR</b> le texte, saisissez <b>n</b> le message d'erreur suivant. Tapez <b>N</b> pour le message d'erreur précédent. |          |
| <sup>2</sup> Après avoir <b>? ERROR</b> tapé, saisissez <b>n</b> le message d'erreur suivant ou le message d'erreur précédent. <b>N</b>                    |          |

Vous pouvez également taper la commande suivante pour ouvrir un fichier journal de CodeDeploy scripts :

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/scripts.log
```

Tapez les commandes suivantes pour parcourir le fichier journal à la recherche des messages d'erreur :

| Commande           | Résultat                                                          |
|--------------------|-------------------------------------------------------------------|
| <b>&amp;stderr</b> | Affiche uniquement les messages d'erreur dans le fichier journal. |
| <b>/stderr</b>     | Recherchez le message d'erreur suivant. <sup>1</sup>              |
| <b>?stderr</b>     | Recherchez le message d'erreur précédent <sup>2</sup> .           |
| <b>G</b>           | Accède à la fin du fichier journal.                               |
| <b>g</b>           | Accède au début du fichier journal.                               |
| <b>q</b>           | Quitte le fichier journal.                                        |
| <b>h</b>           | Donne des informations sur les commandes supplémentaires.         |

<sup>1</sup> Après avoir tapé **/stderr**, saisissez le message d'erreur suivant **n** pour le transfert. Tapez **N** pour le message d'erreur précédent vers l'arrière.

| Commande                                                                                                                                                             | Résultat |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <sup>2</sup> Après avoir tapé <b>stderr</b> , tapez <b>n</b> le message d'erreur suivant en arrière. Tapez <b>N</b> pour le message d'erreur précédent vers l'avant. |          |

## Pour afficher les fichiers journaux de déploiement sur les instances Windows Server

CodeDeploy fichier journal de l'agent : sur les instances Windows Server, le fichier journal de l'agent est stocké à l'emplacement suivant :

C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt

Pour consulter ou analyser le fichier journal de l'agent sur une instance Windows Server, connectez-vous à l'instance, puis tapez la commande suivante pour ouvrir le fichier :

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Pour parcourir le fichier journal à la recherche de messages d'erreur, appuyez sur Ctrl+F, tapez **ERROR** [ ], puis appuyez sur Entrée pour rechercher la première erreur.

CodeDeploy fichiers journaux de scripts : sur les instances Windows Server, les journaux de déploiement sont stockés à l'emplacement suivant :

C:\ProgramData\Amazon\CodeDeploy\*deployment-group-id*\*deployment-id*\logs\scripts.log

Où :

- *deployment-group-id* est une chaîne telle que `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *deployment-id* est un identifiant tel que `d-12EXAMPLE`

Tapez la commande suivante pour ouvrir un fichier journal de CodeDeploy scripts :

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs\scripts.log
```

Pour parcourir le fichier journal à la recherche de messages d'erreur, appuyez sur Ctrl+F, tapez **stderr**, puis appuyez sur Entrée pour rechercher la première erreur.



# Arrêtez un déploiement avec CodeDeploy

Vous pouvez utiliser la CodeDeploy console AWS CLI, le ou le CodeDeploy APIs pour arrêter les déploiements associés à votre AWS compte.

## Warning

L'arrêt d'un déploiement EC2 /On-Premises peut laisser certaines ou toutes les instances de vos groupes de déploiement dans un état de déploiement indéterminé. Pour de plus amples informations, veuillez consulter [Déploiements interrompus ou échoués](#).

Vous pouvez arrêter un déploiement ou arrêter et annuler un déploiement.

- [Arrêt d'un déploiement \(console\)](#)
- [Arrêt d'un déploiement \(interface CLI\)](#)

## Note

Si votre déploiement est un déploiement bleu/vert CloudFormation, vous ne pouvez pas effectuer cette tâche dans la CodeDeploy console. Accédez à la CloudFormation console pour effectuer cette tâche.

## Arrêt d'un déploiement (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

## Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Deployments.

**Note**

Si aucune entrée n'est affichée, assurez-vous que la bonne région est sélectionnée. Dans la barre de navigation, dans le sélecteur de région, choisissez l'une des régions répertoriées dans [Région et Endpoints](#) dans le. Références générales AWS CodeDeploy est pris en charge dans ces régions uniquement.

3. Choisissez le déploiement à arrêter effectuez l'une des opérations suivantes :

1. Choisissez Arrêter le déploiement pour arrêter le déploiement sans restauration.
2. Choisissez Arrêter et restaurer le déploiement pour arrêter et restaurer le déploiement.

Pour de plus amples informations, veuillez consulter [Redéployez et annulez un déploiement avec CodeDeploy](#).

**Note**

Si les options Arrêter le déploiement et Arrêter et restaurer le déploiement sont indisponibles, le déploiement a progressé jusqu'à un point où il ne peut plus être arrêté.

## Arrêt d'un déploiement (interface CLI)

Appelez la commande [stop-deployment](#) en spécifiant l'ID de déploiement. Pour afficher la liste des déploiements IDs, appelez la commande [list-deployments](#).

## Redéployez et annulez un déploiement avec CodeDeploy

CodeDeploy annule les déploiements en redéployant une version précédemment déployée d'une application en tant que nouveau déploiement. Ces déploiements annulés sont des déploiements techniquement nouveaux, avec un nouveau déploiement IDs, plutôt que des versions restaurées d'un déploiement précédent.

Les déploiements peuvent être restaurés automatiquement ou manuellement.

### Rubriques

- [Annulations automatiques](#)

- [Annulations manuelles](#)
- [Flux de travail de restauration et de redéploiement](#)
- [Comportement d'annulation avec du contenu existant](#)

## Annulations automatiques

Vous pouvez configurer un groupe de déploiement ou un déploiement pour être restaurés automatiquement lorsqu'un déploiement échoue, ou lorsqu'un seuil de surveillance que vous spécifiez est atteint. Dans ce cas, la dernière version correcte connue d'une révision de l'application est déployée. Vous configurez des restaurations automatiques lorsque vous créez une application ou créez ou mettez à jour un groupe de déploiement.

Lorsque vous créez un nouveau déploiement, vous pouvez également choisir de remplacer la configuration de la restauration automatique qui a été spécifiée pour le groupe de déploiement.

### Note

Vous pouvez utiliser Amazon Simple Notification Service pour recevoir une notification chaque fois qu'un déploiement est annulé automatiquement. Pour plus d'informations, consultez [Monitoring Deployments with Amazon SNS Event Notifications](#).

Pour plus d'informations sur la configuration de la restauration automatique, consultez [Configuration des options avancées d'un groupe de déploiement](#).

## Annulations manuelles

Si vous n'avez pas configuré les restaurations automatiques, vous pouvez annuler manuellement un déploiement en créant un nouveau déploiement qui utilise une révision d'application précédemment déployée et en suivant les étapes pour redéploier une révision. Vous pouvez le faire si une application a atteint un état inconnu. Plutôt que de passer beaucoup de temps à résoudre des problèmes, vous pouvez redéploier l'application dans un état de fonctionnement connu. Pour de plus amples informations, veuillez consulter [Créez un déploiement avec CodeDeploy](#).

 Note

Si vous supprimez une instance d'un groupe de déploiement, CodeDeploy cela ne désinstalle aucun élément susceptible d'avoir déjà été installé sur cette instance.

## Flux de travail de restauration et de redéploiement

Lorsque la restauration automatique est lancée, ou lorsque vous lancez manuellement un redéploiement ou une annulation manuelle, essayez d'abord de supprimer de chaque instance participante tous les fichiers qui ont été installés avec succès pour la dernière fois. CodeDeploy pour ce faire, vérifiez le fichier de nettoyage :

`/opt/codedeploy-agent/deployment-root/deployment-instructions/deployment-group-ID-cleanup`fichier (pour les instances Amazon Linux, Ubuntu Server et RHEL)

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\deployment-group-ID-cleanup`fichier (pour les instances Windows Server)

S'il existe, CodeDeploy utilise le fichier de nettoyage pour supprimer de l'instance tous les fichiers répertoriés avant de démarrer le nouveau déploiement.

Par exemple, les deux premiers fichiers texte et les deux fichiers de script ont déjà été déployés sur une instance Amazon EC2 exécutant Windows Server, et les scripts ont créé deux autres fichiers texte lors des événements du cycle de vie du déploiement :

```
c:\temp\a.txt (previously deployed by CodeDeploy)
c:\temp\b.txt (previously deployed by CodeDeploy)
c:\temp\c.bat (previously deployed by CodeDeploy)
c:\temp\d.bat (previously deployed by CodeDeploy)
c:\temp\e.txt (previously created by c.bat)
c:\temp\f.txt (previously created by d.bat)
```

Le fichier de nettoyage répertorie uniquement les deux premiers fichiers texte et deux fichiers de script :

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
```

```
c:\temp\d.bat
```

Avant le nouveau déploiement, seuls les deux premiers fichiers texte et les deux fichiers de script CodeDeploy seront supprimés, les deux derniers étant conservés :

```
c:\temp\a.txt will be removed
c:\temp\b.txt will be removed
c:\temp\c.bat will be removed
c:\temp\d.bat will be removed
c:\temp\e.txt will remain
c:\temp\f.txt will remain
```

Dans le cadre de ce processus, je n' CodeDeploy essaierai pas d'annuler ou de réconcilier les actions entreprises par des scripts lors de déploiements précédents lors de redéploiements ultérieurs, qu'il s'agisse de annulations manuelles ou automatiques. Par exemple, si les d.bat fichiers c.bat and contiennent une logique empêchant de recréer les f.txt fichiers e.txt and s'ils existent déjà, les anciennes versions de e.txt et f.txt resteront inchangées lors des CodeDeploy exécutions c.bat et des d.bat déploiements ultérieurs. Vous pouvez ajouter une logique à c.bat et d.bat pour toujours vérifier et supprimer les anciennes versions de e.txt et f.txt avant d'en créer de nouvelles.

## Comportement d'annulation avec du contenu existant

Dans le cadre du processus de déploiement, l' CodeDeploy agent supprime de chaque instance tous les fichiers installés par le déploiement le plus récent. Si des fichiers qui ne faisaient pas partie d'un déploiement précédent apparaissent dans les emplacements de déploiement cibles, vous pouvez choisir ce qu' CodeDeploy il en sera fait lors du prochain déploiement :

- Échec du déploiement : une erreur est signalée et le statut du déploiement passe à Échec.
- Remplacer le contenu : la version du fichier issue de la révision de l'application remplace la version déjà présente sur l'instance.
- Conserver le contenu : le fichier à l'emplacement cible est conservé et la version de la révision de l'application n'est pas copiée sur l'instance.

Vous pouvez choisir ce comportement lorsque vous créez un déploiement. Si vous créez un déploiement dans la console, consultez [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#). Si vous créez un déploiement avec le AWS CLI, voir [Création d'un déploiement de plate-forme de calcul EC2/sur site \(CLI\)](#).

Vous pouvez choisir de conserver les fichiers dont vous voulez qu'ils fassent partie du prochain déploiement sans avoir à les ajouter au package de la révision de l'application. Par exemple, vous pouvez charger directement sur l'instance des fichiers qui sont requis pour le déploiement, mais qui n'ont pas été ajoutés au groupe de révisions de l'application. Vous pouvez également télécharger des fichiers sur l'instance si vos applications se trouvent déjà dans votre environnement de production mais que vous souhaitez les utiliser CodeDeploy pour la première fois pour les déployer.

Dans le cas d'une restauration, où la dernière révision d'application déployée avec succès est redéployée en raison d'une défaillance de déploiement, l'option content-handling de ce dernier déploiement réussi est appliquée au déploiement de la restauration.

Cependant, si le déploiement qui a échoué a été configuré pour remplacer les fichiers au lieu de les conserver, un résultat inattendu peut se produire pendant la restauration. En particulier, les fichiers que vous avez prévu de conserver peuvent être supprimés par l'échec du déploiement. Les fichiers ne sont pas sur l'instance lorsque le déploiement de la restauration s'exécute.

Dans l'exemple suivant, il existe trois déploiements. Tout fichier écrasé (supprimé) au cours du deuxième déploiement défaillant n'est plus disponible (ne peut pas être conservé) lorsque la révision d'application 1 est déployée à nouveau pendant le déploiement 3 :

| Déploiement   | Révision de l'application | Option de remplacement du contenu | Statut du déploiement | Comportement et résultat                                                                                                                                                                            |
|---------------|---------------------------|-----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| déploiement 1 | révision d'application 1  | RETAIN                            | Réussi(e)             | CodeDeploy détecte les fichiers dans les emplacements cibles qui n'ont pas été déployés lors du déploiement précédent. Ces fichiers peuvent être placés à l'intentionnellement pour faire partie du |

| Déploiement   | Révision de l'application | Option de remplacement du contenu | Statut du déploiement | Comportement et résultat                                                                                                                                                                                                                                                             |
|---------------|---------------------------|-----------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                           |                                   |                       | déploiement actuel. Ils sont conservés et enregistrés dans le cadre du package de déploiement actuel.                                                                                                                                                                                |
| déploiement 2 | révision d'application 2  | OVERWRITE                         | Échec                 | <p>Au cours du processus de déploiement, CodeDeploy supprime tous les fichiers faisant partie du déploiement réussi précédent . Cela inclut les fichiers qui ont été conservés pendant le déploiement 1.</p> <p>Cependant, le déploiement échoue pour des raisons indépendantes.</p> |

| Déploiement   | Révision de l'application | Option de remplacement du contenu | Statut du déploiement | Comportement et résultat                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|---------------------------|-----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| déploiement 3 | révision d'application 1  | RETAIN                            |                       | <p>Dans la mesure où l'annulation automatique est activée pour le déploiement ou le groupe de déploiement, CodeDeploy déploie la dernière révision d'application connue, la révision d'application 1.</p> <p>Cependant, les fichiers que vous souhaitez conserver dans le déploiement 1 ont été supprimés avant l'échec du déploiement 2 et ne peuvent pas être récupérés par AWS CodeDeploy. Vous pouvez les ajouter à l'instance vous-même s'ils sont obligatoirement</p> |



| Déploiement | Révision de l'application | Option de remplacement du contenu | Statut du déploiement | Comportement et résultat                                                                        |
|-------------|---------------------------|-----------------------------------|-----------------------|-------------------------------------------------------------------------------------------------|
|             |                           |                                   |                       | res pour la révision d'application 1, ou vous pouvez créer une nouvelle révision d'application. |

## Déployer une application dans un autre AWS compte

Organisations disposent généralement de plusieurs AWS comptes qu'elles utilisent à des fins différentes (par exemple, un pour les tâches d'administration système et un autre pour les tâches de développement, de test et de production, ou un associé aux environnements de développement et de test et un autre associé à l'environnement de production).

Bien que vous puissiez effectuer des tâches connexes dans différents comptes, les groupes de CodeDeploy déploiement et les instances Amazon EC2 sur lesquelles ils sont déployés sont strictement liés aux comptes sous lesquels ils ont été créés. Vous ne pouvez pas, par exemple, ajouter une instance que vous avez lancée dans un compte à un groupe de déploiement dans un autre.

Supposons que vous ayez deux AWS comptes : votre compte de développement et votre compte de production. Vous travaillez principalement dans le compte de développement, mais vous souhaitez être capable de lancer des déploiements dans votre compte de production sans un ensemble complet d'informations d'identification ou sans avoir à vous déconnecter du compte de développement pour vous connecter au compte de production.

Après avoir suivi les étapes de configuration entre comptes, vous pouvez lancer des déploiements appartenant à un autre compte de votre organisation, sans avoir besoin d'un ensemble complet d'informations d'identification pour cet autre compte. Pour ce faire, vous utilisez en partie une fonctionnalité fournie par le AWS Security Token Service (AWS STS) qui vous accorde un accès temporaire à ce compte.

## Étape 1 : créer un compartiment S3 dans l'un ou l'autre des comptes

Dans le compte de développement ou le compte de production :

- Si ce n'est pas déjà fait, créez un compartiment Amazon S3 dans lequel les révisions de l'application pour le compte de production seront stockées. Pour plus d'informations, consultez [Créer un compartiment dans Amazon S3](#). Vous pouvez même utiliser les mêmes révisions d'application et compartiment pour les deux comptes, en déployant les mêmes fichiers dans votre environnement de production que vous avez testés et vérifiés dans votre compte de développement.

## Étape 2 : accorder des autorisations de compartiment Amazon S3 au profil d'instance IAM du compte de production

Si le compartiment Amazon S3 que vous avez créé à l'étape 1 se trouve dans votre compte de production, cette étape n'est pas obligatoire. Le rôle que vous assumerez plus tard aura déjà accès à ce compartiment, car il figure également dans le compte de production.

Si vous avez créé le compartiment Amazon S3 dans le compte de développement, procédez comme suit :

- Dans le compte de production, créez un profil d'instance IAM. Pour plus d'informations, consultez [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).

### Note

Notez l'ARN de ce profil d'instance IAM. Vous devez l'ajouter à la stratégie entre compartiments que vous créez ensuite.

- Dans le compte de développement, donnez accès au bucket Amazon S3 que vous avez créé dans le compte de développement au profil d'instance IAM que vous venez de créer dans votre compte de production. Pour plus d'informations, voir [Exemple 2 : le propriétaire du bucket accorde des autorisations de bucket entre comptes](#).

Notez les éléments suivants lorsque vous terminez le processus d'octroi des autorisations de compartiment entre comptes :

- Dans cet exemple de procédure, le compte A représente votre compte de développement et le compte B représente votre compte de production.

- Lorsque vous [effectuez les tâches du compte A \(compte de développement\)](#), modifiez la stratégie de compartiment suivante pour accorder des autorisations entre comptes au lieu d'utiliser l'exemple de stratégie fourni dans la procédure pas à pas.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "Cross-account permissions",
 "Effect": "Allow",
 "Principal": {
 "AWS": "arn:aws:iam::111122223333:role/role-name"
 },
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::bucket-name/*"
]
 }
]
}
```

**account-id** représente le numéro de compte du compte de production sur lequel vous venez de créer le profil d'instance IAM.

**role-name** représente le nom du profil d'instance IAM que vous venez de créer.

**bucket-name** représente le nom du compartiment que vous avez créé à l'étape 1. N'oubliez pas d'inclure `/*` après le nom de votre compartiment pour accorder l'accès à chacun des fichiers à l'intérieur du compartiment.

## Étape 3 : Création de ressources et d'un rôle multicompte dans le compte de production

Dans votre compte de production :

- Créez vos CodeDeploy ressources (application, groupe de déploiement, configuration de déploiement, instances Amazon EC2, profil d'instance Amazon EC2, rôle de service, etc.) en suivant les instructions de ce guide.
- Créez un rôle supplémentaire, un rôle IAM entre comptes, qu'un utilisateur de votre compte de développement peut assumer pour effectuer CodeDeploy des opérations dans ce compte de production.

Utilisez la [procédure pas à pas : déléguez l'accès entre les AWS comptes à l'aide des rôles IAM](#) comme guide pour vous aider à créer le rôle entre comptes. Au lieu d'ajouter les exemples d'autorisations dans la procédure pas à pas à votre document de politique, vous devez associer au moins les deux politiques AWS fournies suivantes au rôle :

- `AmazonS3FullAccess` : requise uniquement si le compartiment S3 figure dans le compte de développement. Fournit au rôle de compte de production assumé un accès complet aux services et ressources Amazon S3 dans le compte de développement, où la révision est stockée.
- `AWSCodeDeployDeployerAccess`: Permet à un utilisateur d'enregistrer et de déployer des révisions.

Si vous souhaitez créer et gérer des groupes de déploiement et non pas seulement initier des déploiements, ajoutez la stratégie `AWSCodeDeployFullAccess` à la place de la stratégie `AWSCodeDeployDeployerAccess`. Pour plus d'informations sur l'utilisation des politiques gérées par IAM pour accorder des autorisations pour les CodeDeploy tâches, consultez [AWS politiques gérées \(prédéfinies\) pour CodeDeploy](#).

Vous pouvez attacher des stratégies supplémentaires si vous souhaitez effectuer des tâches dans d'autres services AWS tout en utilisant ce rôle entre comptes.

#### Important

Lorsque vous créez le rôle IAM entre comptes, notez les informations dont vous aurez besoin pour accéder au compte de production.

Pour utiliser le AWS Management Console pour changer de rôle, vous devez fournir l'un des éléments suivants :

- Une URL pour accéder au compte de production avec les informations d'identification du rôle assumé. Vous trouverez cette URL sur la page Révision, qui s'affiche à la fin du processus de création de rôle entre comptes.
- Le nom du rôle entre comptes et l'alias ou l'ID du compte.

Pour utiliser le AWS CLI pour changer de rôle, vous devez fournir les éléments suivants :

- Le nom ARN du rôle entre comptes que vous allez assumer.

## Étape 4 : télécharger la révision de l'application dans le compartiment Amazon S3

Dans le compte dans lequel vous avez créé le compartiment Amazon S3 :

- Téléchargez la révision de votre application dans le compartiment Amazon S3. Pour plus d'informations, consultez [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#).

## Étape 5 : assumer le rôle multi-comptes et déployer des applications

Dans le compte de développement, vous pouvez utiliser le AWS CLI ou le AWS Management Console pour assumer le rôle entre comptes et lancer le déploiement dans le compte de production.

Pour obtenir des instructions sur la façon d'utiliser le AWS Management Console pour changer de rôle et lancer des déploiements, voir [Basculer vers un rôle \(AWS Management Console\)](#) et [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#).

Pour savoir comment utiliser le AWS CLI pour assumer le rôle entre comptes et lancer des déploiements, voir [Passer à un rôle IAM \(AWS Command Line Interface\)](#) et [Création d'un déploiement de plate-forme de calcul EC2/sur site \(CLI\)](#).

Pour plus d'informations sur l'attribution d'un rôle AWS STS, consultez [AssumeRole](#) le [guide de l'AWS Security Token Service utilisateur](#) et [assume-role](#) dans le manuel de référence des [AWS CLI commandes](#).

Voir aussi :

- [CodeDeploy: Déploiement d'un compte de développement vers un compte de production](#)

# Utiliser l' CodeDeploy agent pour valider un package de déploiement sur une machine locale

À l'aide de l' CodeDeploy agent, vous pouvez déployer du contenu sur une instance à laquelle vous êtes connecté. Cela vous permet de tester l'intégrité d'un fichier de spécification d'application (AppSpec fichier) que vous souhaitez utiliser dans le cadre d'un déploiement et du contenu que vous souhaitez déployer.

Il n'est pas nécessaire de créer une application et un groupe de déploiement. Si vous souhaitez déployer du contenu stocké sur l'instance locale, vous n'avez même pas besoin de AWS compte. Pour les tests les plus simples, vous pouvez exécuter la `codedeploy-local` commande, sans spécifier d'options, dans un répertoire contenant le AppSpec fichier et le contenu à déployer. L'outil inclut aussi des options pour d'autres scénarios de test.

En validant un package de déploiement sur un ordinateur local, vous pouvez :

- Tester l'intégrité d'une révision d'application.
- Testez le contenu d'un AppSpec fichier.
- Essayez CodeDeploy pour la première fois avec le code de votre application existant.
- Déployer rapidement du contenu lorsque vous êtes déjà connecté à une instance.

Vous pouvez utiliser le contenu de déploiement stocké sur l'instance locale ou dans un type de référentiel distant pris en charge (compartiments Amazon S3 ou GitHub référentiels publics).

## Conditions préalables

Avant de lancer un déploiement local, effectuez les étapes suivantes :

- Créez ou utilisez un type d'instance pris en charge par l' CodeDeploy agent. Pour plus d'informations, consultez [Systèmes d'exploitation pris en charge par l' CodeDeploy agent](#).
- Installez la version 1.0.1.1352 ou ultérieure de l'agent. CodeDeploy Pour plus d'informations, consultez [Installation de l' CodeDeploy agent](#).
- Si vous déployez votre contenu à partir d'un compartiment ou d'un GitHub référentiel Amazon S3, configurez un utilisateur avec lequel vous pourrez l'utiliser CodeDeploy. Pour plus d'informations, consultez [Étape 1 : Configuration](#).
- Si vous déployez la révision de votre application à partir d'un compartiment Amazon S3, créez un compartiment Amazon S3 dans la région dans laquelle vous travaillez et appliquez une politique

de compartiment Amazon S3 au compartiment. Cette stratégie accorde à vos instances les autorisations requises pour télécharger la révision d'application.

Par exemple, la politique de compartiment Amazon S3 suivante permet à toute instance Amazon EC2 associée à un profil d'instance IAM contenant l'ARN de `arn:aws:iam::444455556666:role/CodeDeployDemo` télécharger n'importe où dans le compartiment Amazon S3 nommé : `amzn-s3-demo-bucket`

```
{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:role/CodeDeployDemo"
]
 }
 }
]
}
```

La politique de compartiment Amazon S3 suivante permet à toute instance sur site associée à un utilisateur IAM contenant l'ARN de `arn:aws:iam::444455556666:user/CodeDeployUser` télécharger depuis n'importe quel emplacement du compartiment Amazon S3 nommé : `amzn-s3-demo-bucket`

```
{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
 "Principal": {
```

```
 "AWS": [
 "arn:aws:iam::444455556666:user/CodeDeployUser"
]
 }
}
```

Pour plus d'informations sur la façon de générer et d'associer une politique de compartiment Amazon S3, consultez les [exemples de politique de compartiment](#).

- Si vous déployez la révision de votre application à partir d'un compartiment ou d'un GitHub référentiel Amazon S3, configurez un profil d'instance IAM et attachez-le à l'instance. Pour plus d'informations, consultez [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#), [Créez une instance Amazon EC2 pour CodeDeploy \(AWS CLI ou une console Amazon EC2\)](#) et [Création d'une instance Amazon EC2 pour CodeDeploy \(modèle\)CloudFormation](#).
- Si vous déployez votre contenu depuis GitHub, créez un GitHub compte et un dépôt public. Pour créer un GitHub compte, consultez la section [S'inscrire GitHub](#). Pour créer un GitHub dépôt, voir [Création d'un dépôt](#).

#### Note

Les référentiels privés ne sont pas pris en charge pour l'instant. Si votre contenu est stocké dans un GitHub référentiel privé, vous pouvez le télécharger sur l'instance et utiliser l'option `bundle-location` pour spécifier son chemin local.

- Préparez le contenu (y compris un AppSpec fichier) que vous souhaitez déployer sur l'instance et placez-le sur l'instance locale, dans votre compartiment Amazon S3 ou dans votre GitHub référentiel. Pour plus d'informations, consultez [Travailler avec les révisions d'applications pour CodeDeploy](#).
- Si vous souhaitez utiliser des valeurs autres que les valeurs par défaut pour d'autres options de configuration, créez le fichier de configuration et placez-le sur l'instance (`/etc/codedeploy-agent/conf/codedeployagent.yml` pour les instances Amazon Linux, RHEL ou Ubuntu Server ou `C:\ProgramData\Amazon\CodeDeploy\conf.yml` pour les instances Windows Server). Pour plus d'informations, consultez [CodeDeploy référence de configuration de l'agent](#).



 Note

Si vous utilisez un fichier de configuration sur des instances Amazon Linux, RHEL ou Ubuntu Server, vous devez soit :

- Utiliser les variables `:root_dir:` et `:log_dir:` pour spécifier des emplacements autres que ceux par défaut pour les dossiers du répertoire de journaux et du répertoire racine du déploiement.
- `sudo` à utiliser pour exécuter les commandes de CodeDeploy l'agent.

## Création d'un déploiement local

Sur l'instance où vous souhaitez créer le déploiement local, ouvrez une session de terminal (instances Amazon Linux, RHEL ou Ubuntu Server) ou une invite de commande (Windows Server) pour exécuter les commandes de l'outil.

 Note

La commande `codedeploy-local` est installée dans les emplacements suivants :

- Sur Amazon Linux, RHEL ou Ubuntu Server : `/opt/codedeploy-agent/bin`
- Sur Windows Server : `C:\ProgramData\Amazon\CodeDeploy\bin`.

## Syntaxe de base des commandes

```
codedeploy-local [options]
```

## Résumé

```
codedeploy-local
[--bundle-location <value>]
[--type <value>]
[--file-exists-behavior <value>]
[--deployment-group <value>]
[--events <comma-separated values>]
[--agent-configuration-file <value>]
```

```
[--appspec-filename <value>]
```

## Options

**-l, --bundle-location**

Emplacement du bundle de la révision d'application. Si vous ne spécifiez pas d'emplacement, l'outil utilise par défaut le répertoire dans lequel vous travaillez actuellement. Si vous spécifiez une valeur pour `--bundle-location`, vous devez également en spécifier une pour `--type`.

Exemples de format pour l'emplacement du bundle :

- Instance locale de serveur Amazon Linux, RHEL ou Ubuntu : `/path/to/local/bundle.tgz`
- Instance Windows Server locale : `C:/path/to/local/bundle`
- Compartiment Amazon S3 : `s3://amzn-s3-demo-bucket/bundle.tar`
- GitHub référentiel : `https://github.com/account-name/repository-name/`

**-t, --type**

Format du bundle de la révision d'application. Les types pris en charge incluent `tgz`, `tar`, `zip` et `directory`. Si vous ne spécifiez aucun type, l'outil utilise `directory` par défaut. Si vous spécifiez une valeur pour `--type`, vous devez également en spécifier une pour `--bundle-location`.

**-b, --file-exists-behavior**

Indique le traitement à appliquer aux fichiers qui existent déjà dans un emplacement cible du déploiement mais qui ne faisaient pas partie d'un précédent déploiement réussi. Les options incluent `DISALLOW`, `OVERWRITE`, `RETAIN`. Pour plus d'informations, consultez [fileExistsBehavior](#) la section [Référence des AWS CodeDeploy API](#).

**-g, --deployment-group**

Chemin du dossier constituant l'emplacement cible du contenu à déployer. Si vous ne spécifiez aucun dossier, l'outil en crée un nommé `default-local-deployment-group` dans le répertoire racine de votre déploiement. Pour chaque déploiement local que vous créez, l'outil crée un sous-répertoire à l'intérieur de ce dossier avec un nom tel que `d-98761234-local`.

**-e, --events**

Un ensemble de hooks d'événements de remplacement du cycle de vie que vous souhaitez exécuter, dans l'ordre, à la place des événements que vous avez répertoriés dans le AppSpec fichier. Plusieurs hooks peuvent être spécifiés, séparés par des virgules. Vous pouvez utiliser cette option si :

- Vous souhaitez exécuter un ensemble d'événements différent sans avoir à mettre à jour le AppSpec fichier.
- Vous souhaitez exécuter un hook d'événement unique en tant qu'exception au contenu du AppSpec fichier, par exemple `ApplicationStop`.

Si vous ne spécifiez ni n'installez `DownloadBundle` les événements dans la liste de remplacement, ils s'exécuteront avant tous les hooks d'événements que vous spécifiez. Si vous incluez `DownloadBundle` et `Installer` dans la liste des `--events` options, elles doivent être précédées uniquement des événements qui s'exécutent normalement avant elles lors CodeDeploy des déploiements. Pour plus d'informations, consultez [AppSpec section « crochets »](#).

`-c, --agent-configuration-file`

Emplacement d'un fichier de configuration à utiliser pour le déploiement, si vous le stockez à un emplacement autre que celui par défaut. Un fichier de configuration spécifie des options alternatives pour les autres valeurs et comportements par défaut dans le cadre d'un déploiement.

Par défaut, les fichiers de configuration sont stockés dans `/etc/codedeploy-agent/conf/codedeployagent.yml` (instances Amazon Linux, RHEL ou Ubuntu Server) ou `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server). Pour de plus amples informations, veuillez consulter [CodeDeploy référence de configuration de l'agent](#).

`-A, --appspec-filename`

Le nom du AppSpec fichier. Pour les déploiements locaux, les valeurs acceptées sont `appspec.yml` et `appspec.yaml`. Par défaut, le AppSpec fichier est appelé `appspec.yml`.

`-h, --help`

Affiche un résumé du contenu d'aide.

`-v, --version`

Affiche le numéro de version de l'outil.

## Exemples

Voici des exemples de formats de commande valides.

```
codedeploy-local
```

```
codedeploy-local --bundle-location /path/to/local/bundle/directory
```

```
codedeploy-local --bundle-location C:/path/to/local/bundle.zip --type zip --deployment-group my-deployment-group
```

```
codedeploy-local --bundle-location /path/to/local/directory --type directory --deployment-group my-deployment-group
```

Déployez un bundle depuis Amazon S3 :

```
codedeploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.tgz --type tgz
```

```
codedeploy-local --bundle-location s3://amzn-s3-demo-bucket/bundle.zip?versionId=1234&etag=47e8 --type zip --deployment-group my-deployment-group
```

Déployez un bundle à partir d'un GitHub dépôt public :

```
codedeploy-local --bundle-location https://github.com/awslabs/aws-codedeploy-sample-tomcat --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/master --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/HEAD --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/1a2b3c4d --type zip
```

Déploiement d'un bundle en spécifiant plusieurs événements de cycle de vie :

```
codedeploy-local --bundle-location /path/to/local/bundle.tar --type tar --application-
folder my-deployment --events DownloadBundle,Install,ApplicationStart,HealthCheck
```

Arrêtez une application précédemment déployée à l'aide de l'événement `ApplicationStop` du cycle de vie :

```
codedeploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --deployment-
group --events ApplicationStop
```

Déploiement à l'aide d'un ID de groupe de déploiement spécifique :

```
codedeploy-local --bundle-location C:/path/to/local/bundle/directory --deployment-group
1234abcd-5dd1-4774-89c6-30b107ac5dca
```

```
codedeploy-local --bundle-location C:/path/to/local/bundle.zip --type zip --deployment-
group 1234abcd-5dd1-4774-89c6-30b107ac5dca
```

# Surveillance des déploiements dans CodeDeploy

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité CodeDeploy 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. Avant de commencer la surveillance CodeDeploy, vous devez toutefois créer un plan de surveillance comprenant des réponses aux questions suivantes :

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

L'étape suivante consiste à établir une base de référence pour des CodeDeploy performances normales dans votre environnement, en mesurant les performances à différents moments et dans différentes conditions de charge. Pendant que vous surveillez CodeDeploy, stockez les données de surveillance historiques afin de pouvoir les comparer aux données de performance actuelles, d'identifier les modèles de performance normaux et les anomalies de performance, et de concevoir des méthodes pour résoudre les problèmes.

Par exemple, si vous utilisez CodeDeploy, vous pouvez surveiller l'état des déploiements et des instances cibles. Lorsque les déploiements ou les instances échouent, vous devrez peut-être reconfigurer un fichier de spécifications d'application, réinstaller ou mettre à jour l' CodeDeploy agent, mettre à jour les paramètres d'une application ou d'un groupe de déploiement, ou apporter des modifications aux paramètres d'instance ou à un AppSpec fichier.

Pour établir une référence, vous devez, au Moins, superviser les éléments suivants :

- Événements et état de déploiement
- Événements et état d'instance

# Outils de surveillance automatique

AWS fournit divers outils que vous pouvez utiliser pour surveiller CodeDeploy. Vous pouvez configurer certains outils pour qu'ils effectuent la supervision automatiquement, tandis que d'autres nécessitent une intervention manuelle. Nous vous recommandons d'automatiser le plus possible les tâches de supervision.

Vous pouvez utiliser les outils de surveillance automatique suivants pour surveiller CodeDeploy et signaler tout problème :

- Amazon CloudWatch Alarms : surveillez une seule métrique sur une période que vous spécifiez et effectuez une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. L'action est une notification envoyée à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une politique Amazon EC2 Auto Scaling. CloudWatch les alarmes n'appellent pas d'actions simplement parce qu'elles sont dans un état particulier ; l'état doit avoir changé et être maintenu pendant un certain nombre de périodes. Pour de plus amples informations, veuillez consulter [Monitoring Deployments with Amazon CloudWatch Tools](#).

Pour plus d'informations sur la mise à jour de votre rôle de service pour utiliser la surveillance d'alarmes CloudWatch , consultez la rubrique [Accorder CloudWatch des autorisations à un rôle CodeDeploy de service](#). Pour plus d'informations sur l'ajout de la surveillance des CloudWatch alarmes à vos CodeDeploy opérations [Créez une application avec CodeDeploy](#), consultez [Créez un groupe de déploiement avec CodeDeploy](#), ou [Modifiez les paramètres du groupe de déploiement avec CodeDeploy](#).

- Amazon CloudWatch Logs — Surveillez, stockez et accédez à vos fichiers journaux depuis AWS CloudTrail ou d'autres sources. Pour plus d'informations, consultez la section [Monitoring Log Files](#) dans le guide de CloudWatch l'utilisateur Amazon.

Pour plus d'informations sur l'utilisation de la CloudWatch console pour afficher CodeDeploy les journaux, voir [Afficher CodeDeploy les CloudWatch journaux dans la console Logs](#).

- Amazon CloudWatch Events : associez les événements et acheminez-les vers une ou plusieurs fonctions ou flux cibles afin d'apporter des modifications, de recueillir des informations d'état et de prendre des mesures correctives. Pour plus d'informations, consultez la section [Qu'est-ce qu'Amazon CloudWatch Events](#) dans le guide de CloudWatch l'utilisateur Amazon.

Pour plus d'informations sur l'utilisation CloudWatch des événements dans vos CodeDeploy opérations, consultez [Surveillance des déploiements avec Amazon CloudWatch Events](#).

- AWS CloudTrail Surveillance des journaux : partagez les fichiers journaux entre les comptes, surveillez les fichiers CloudTrail CloudWatch journaux en temps réel en les envoyant à Logs, écrivez des applications de traitement des journaux en Java et vérifiez que vos fichiers journaux n'ont pas changé après leur livraison par CloudTrail. Pour plus d'informations, consultez la section [Utilisation des fichiers CloudTrail journaux](#) dans le guide de AWS CloudTrail l'utilisateur.

Pour plus d'informations sur l'utilisation CloudTrail avec CodeDeploy, voir [Monitoring Deployments](#).

- Amazon Simple Notification Service : configurez des déclencheurs pilotés par des événements pour recevoir des notifications par SMS ou par e-mail concernant le déploiement et les événements liés à l'instance, tels que le succès ou l'échec. Pour plus d'informations, consultez [Créer un sujet](#) et [Qu'est-ce qu'Amazon Simple Notification Service](#) ?

Pour plus d'informations sur la configuration des notifications Amazon SNS pour CodeDeploy, consultez. [Monitoring Deployments with Amazon SNS Event Notifications](#)

## Outils de surveillance manuelle

Une autre partie importante de la surveillance CodeDeploy consiste à surveiller manuellement les éléments non couverts par les CloudWatch alarmes. Le CodeDeploy tableau de bord de AWS console et les autres tableaux de bord fournissent une at-a-glance vue de l'état de votre AWS environnement. CloudWatch Nous vous recommandons de consulter également les fichiers journaux des CodeDeploy déploiements.

- CodeDeploy la console affiche :
  - L'état des déploiements
  - La date et l'heure de chaque dernière tentative et du dernier déploiement réussi d'une révision
  - Le nombre d'instances qui ont réussi, échoué, ont été ignorées ou sont en cours dans le cadre d'un déploiement
  - L'état des instances sur site
  - La date et l'heure de l'enregistrement ou de l'annulation de l'enregistrement des instances sur site
- CloudWatch la page d'accueil montre :
  - Alarmes et statuts en cours



- Graphiques des alarmes et des ressources
- Statut d'intégrité du service

En outre, vous pouvez utiliser CloudWatch pour effectuer les opérations suivantes :

- Créer des [tableaux de bord personnalisés](#) pour surveiller les services de votre choix
- Représenter graphiquement les données de métriques pour résoudre les problèmes et découvrir les tendances
- Recherchez et parcourez tous les indicateurs de vos AWS ressources
- Créer et modifier des alarmes pour être informé des problèmes

## Rubriques

- [Monitoring Deployments with Amazon CloudWatch Tools](#)
- [Monitoring Deployments](#)
- [Monitoring Deployments with Amazon SNS Event Notifications](#)

# Surveillance des déploiements à l'aide des outils Amazon CloudWatch

Vous pouvez surveiller CodeDeploy les déploiements à l'aide CloudWatch des outils suivants : Amazon CloudWatch Events, CloudWatch alarmes et Amazon CloudWatch Logs.

L'examen des journaux créés par l' CodeDeploy agent et des déploiements peut vous aider à résoudre les causes des échecs de déploiement. Au lieu de consulter les CodeDeploy journaux sur une instance à la fois, vous pouvez utiliser CloudWatch Logs pour surveiller tous les journaux dans un emplacement central.

Pour plus d'informations sur l'utilisation des CloudWatch alarmes et CloudWatch des événements pour surveiller vos CodeDeploy déploiements, consultez les rubriques suivantes.

## Rubriques

- [Surveillance des déploiements avec des CloudWatch alarmes dans CodeDeploy](#)
- [Surveillance des déploiements avec Amazon CloudWatch Events](#)

# Surveillance des déploiements avec des CloudWatch alarmes dans CodeDeploy

Vous pouvez créer une CloudWatch alarme pour une instance ou un groupe Amazon EC2 Auto Scaling que vous utilisez dans le cadre de CodeDeploy vos opérations. Une alarme surveille une seule métrique sur une période que vous spécifiez et exécute une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. CloudWatch les alarmes appellent des actions lorsque leur état change (par exemple, de OK àALARM).

À l'aide de la fonctionnalité CloudWatch d'alarme native, vous pouvez spécifier toutes les actions prises en charge en CloudWatch cas d'échec d'une instance que vous utilisez dans le cadre d'un déploiement, comme l'envoi d'une notification Amazon SNS ou l'arrêt, la résiliation, le redémarrage ou la restauration d'une instance. Pour vos CodeDeploy opérations, vous pouvez configurer un groupe de déploiement pour arrêter un déploiement chaque fois qu'une CloudWatch alarme que vous associez au groupe de déploiement est activée.

Vous pouvez associer jusqu'à dix CloudWatch alarmes à un groupe CodeDeploy de déploiement. Si une des alarmes spécifiées est activée, le déploiement s'arrête et l'état est mis à jour à Stopped. Pour utiliser cette option, vous devez accorder CloudWatch des autorisations à votre rôle CodeDeploy de service.

Pour plus d'informations sur la configuration des CloudWatch alarmes dans la CloudWatch console, consultez la section [Création d' CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon.

Pour plus d'informations sur l'association CloudWatch d'une alarme à un groupe de déploiement dans CodeDeploy, reportez-vous [Créez un groupe de déploiement avec CodeDeploy](#) aux sections et [Modifiez les paramètres du groupe de déploiement avec CodeDeploy](#).

## Rubriques

- [Accorder CloudWatch des autorisations à un rôle CodeDeploy de service](#)

## Accorder CloudWatch des autorisations à un rôle CodeDeploy de service

Avant de pouvoir utiliser la surveillance des CloudWatch alarmes dans le cadre de vos déploiements, le rôle de service que vous utilisez dans le cadre de vos CodeDeploy opérations doit être autorisé à accéder aux CloudWatch ressources.

## Pour accorder CloudWatch des autorisations à un rôle de service

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans la console IAM, dans le volet de navigation, sélectionnez Rôles.
3. Choisissez le nom du rôle de service que vous utilisez dans vos AWS CodeDeploy opérations.
4. Dans l'onglet Autorisations, dans la zone Stratégies en ligne, choisissez Créer une stratégie de rôle.

—ou—

Si le bouton Créer une stratégie de rôle n'est pas disponible, développez la zone Stratégies en ligne, puis choisissez cliquez ici.

5. Sur la page Set Permissions (Définir les autorisations), choisissez Stratégie personnalisée, puis Sélectionner.
6. Sur la page Examiner une stratégie, dans le champ Nom de la stratégie, tapez un nom pour identifier cette stratégie, comme CWA1arms.
7. Collez la stratégie suivante dans le champ Policy Document (Document de stratégie) :

JSON

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

8. Choisissez Apply Policy (Appliquer la stratégie).

## Surveillance des déploiements avec Amazon CloudWatch Events

Vous pouvez utiliser Amazon CloudWatch Events pour détecter et réagir aux modifications de l'état d'une instance ou d'un déploiement (un « événement ») dans le cadre de vos CodeDeploy

opérations. Ensuite, en fonction des règles que vous créez, CloudWatch Events invoquera une ou plusieurs actions cibles lorsqu'un déploiement ou une instance entre dans l'état que vous spécifiez dans une règle. En fonction du type de modification de l'état, vous pouvez envoyer des notifications, capturer les informations d'état, prendre des mesures correctives, déclencher des événements ou prendre d'autres mesures. Vous pouvez sélectionner les types de cibles suivants lorsque vous utilisez CloudWatch des événements dans le cadre de vos CodeDeploy opérations :

- AWS Lambda fonctions
- Flux Kinesis
- Files d'attente Amazon SQS
- Cibles intégrées (EC2 CreateSnapshot API call, EC2 RebootInstances API call, EC2 StopInstances API call, et EC2 TerminateInstances API call)
- Rubriques Amazon SNS

Voici quelques cas d'utilisation :

- Utilisez une fonction Lambda pour transmettre une notification à un canal Slack chaque fois que des déploiements échouent.
- Envoyez des données sur les déploiements ou instances à un flux Kinesis pour prendre en charge la surveillance complète et en temps réel de l'état.
- Utilisez des actions CloudWatch d'alarme pour arrêter, résilier, redémarrer ou récupérer automatiquement les EC2 instances Amazon lorsqu'un événement de déploiement ou d'instance que vous spécifiez se produit.

Le reste de cette rubrique décrit la procédure de base pour créer une règle d' CloudWatch événements pour CodeDeploy. Avant de créer des règles d'événements à utiliser dans le cadre de vos CodeDeploy opérations, vous devez toutefois effectuer les opérations suivantes :

- Complétez les prérequis CloudWatch relatifs aux événements. Pour plus d'informations, consultez les [conditions préalables à Amazon CloudWatch Events](#).
- Familiarisez-vous avec les événements, les règles et les cibles dans CloudWatch Événements. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon CloudWatch Events ?](#) et [nouveaux CloudWatch événements : suivez les modifications apportées à vos AWS ressources et répondez-y](#).
- Créez la ou les cible(s), que vous allez utiliser dans vos règles d'événement.

Pour créer une règle d' CloudWatch événements pour CodeDeploy :

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, sélectionnez Events (Évènements).
3. Choisissez Create rule (Créer une règle) et, sous Event selector (Sélection d'événement), choisissez AWS CodeDeploy.
4. Spécifiez un type de détail :
  - Pour créer une règle qui s'applique à toutes les modifications d'état des instances et des déploiements, choisissez Any detail type (Tout type de détail), puis passez à l'étape 6.
  - Pour créer une règle qui s'applique uniquement aux instances, choisissez Type de détail spécifique, puis choisissez Notification de changement d'état de l'CodeDeploy instance.
  - Pour créer une règle qui s'applique uniquement aux déploiements, choisissez Type de détail spécifique, puis choisissez Notification de changement d'état CodeDeploy du déploiement.
5. Spécifiez les modifications d'état auxquelles la règle s'applique :
  - Pour créer une règle qui s'applique à toutes les modifications d'état, choisissez Any state (Tout état).
  - Pour créer une règle qui s'applique à certaines modifications d'état uniquement, choisissez Specific state(s) (État(s) spécifique(s)), puis sélectionnez une ou plusieurs valeurs d'état dans la liste. Le tableau suivant répertorie les valeurs d'état que vous pouvez choisir :

| Valeurs de l'état de déploiement | Valeurs de l'état d'instance |
|----------------------------------|------------------------------|
| ÉCHEC                            | ÉCHEC                        |
| START                            | START                        |
| ARRÊTER                          | READY                        |
| QUEUED                           | SUCCESS                      |
| READY                            |                              |
| SUCCÈS                           |                              |

6. Spécifiez CodeDeploy les applications auxquelles la règle s'applique :

- Pour créer une règle qui s'applique à toutes les applications, choisissez Any application (Toute application), puis passez à l'étape 8.
  - Pour créer une règle qui s'applique à une seule application, choisissez Specific application (Application spécifique), puis choisissez le nom de l'application dans la liste.
7. Spécifiez les groupes de déploiement auxquels s'applique la règle :
    - Pour créer une règle qui s'applique à tous les groupes de déploiement associés à l'application sélectionnée, choisissez Any deployment group (Tout groupe de déploiement).
    - Pour créer une règle qui s'applique à un seul des groupes de déploiement associés à l'application sélectionnée, choisissez Specific deployment group(s) (Groupe(s) de déploiement spécifique(s)), puis sélectionnez le nom du groupe de déploiement dans la liste.
  8. Passez en revue la configuration de votre règle pour vous assurer qu'elle correspond à vos besoins de surveillance des événements.
  9. Dans la zone Targets, choisissez Add target\*.
  10. Dans la liste Sélectionner un type de cible, choisissez le type de cible que vous avez préparé pour l'utiliser avec cette règle, puis configurez les options supplémentaires requises par ce type de cible.
  11. Choisissez Configure details (Configurer les détails).
  12. Sur la page Configure rule details (Configurer les détails de règle), tapez un nom et une description pour la règle, puis cochez la case État pour activer la règle tout de suite.
  13. Si la règle vous convient, choisissez Créer une règle.

## Surveillance des déploiements avec AWS CloudTrail

CodeDeploy est intégré à CloudTrail un service qui capture les appels d'API effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux CodeDeploy dans un compartiment Amazon S3 que vous spécifiez. CloudTrail capture les appels d'API depuis la CodeDeploy console AWS CLI, depuis des CodeDeploy commandes via ou CodeDeploy APIs directement. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer à quelle demande a été faite CodeDeploy, l'adresse IP source à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite, etc. Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le [Guide de AWS CloudTrail l'utilisateur](#).

## CodeDeploy informations dans CloudTrail

Lorsque la CloudTrail journalisation est activée dans votre AWS compte, les appels d'API effectués aux CodeDeploy actions sont suivis dans des fichiers journaux. CodeDeploy les enregistrements sont écrits avec d'autres enregistrements AWS de service dans un fichier journal. CloudTrail détermine à quel moment créer et écrire dans un nouveau fichier en fonction d'une période et de la taille du fichier.

Toutes les CodeDeploy actions sont enregistrées et documentées dans la [référence de ligne de AWS CodeDeploy commande](#) et la [référence AWS CodeDeploy d'API](#). Par exemple, les appels pour créer des déploiements, supprimer des applications et enregistrer des révisions d'applications génèrent des entrées dans des fichiers CloudTrail journaux.

Chaque entrée du journal contient des informations sur la personne qui a généré la demande. Les informations d'identité utilisateur contenues dans le journal vous aident à déterminer si la demande a été faite avec des informations d'identification root ou utilisateur, avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré, ou par un autre AWS service. Pour plus d'informations, reportez-vous au champ `userIdentity` dans la [Référence des événements CloudTrail](#).

Vous pouvez stocker vos fichiers journaux dans votre compartiment aussi longtemps que vous le souhaitez, mais vous pouvez également définir des règles de cycle de vie Amazon S3 pour archiver ou supprimer automatiquement les fichiers journaux. Par défaut, le chiffrement côté serveur (SSE) Amazon S3 est utilisé pour chiffrer vos fichiers journaux.

Vous pouvez avoir à CloudTrail publier des notifications Amazon SNS lorsque de nouveaux fichiers journaux sont livrés. Pour plus d'informations, consultez [Configuration des notifications Amazon SNS pour CloudTrail](#).

Vous pouvez également agréger les fichiers CodeDeploy journaux de plusieurs AWS régions et de plusieurs AWS comptes dans un seul compartiment Amazon S3. Pour plus d'informations, consultez la section [Réception de fichiers CloudTrail journaux provenant de plusieurs régions](#).

## Comprendre les entrées du fichier CodeDeploy journal

CloudTrail les fichiers journaux peuvent contenir une ou plusieurs entrées de journal, chaque entrée étant composée de plusieurs événements au format JSON. Une entrée de journal représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'action demandée, sur tous les paramètres, sur la date et l'heure de l'action, etc. Les entrées de journal ne

sont garanties dans aucun ordre particulier. Cela signifie qu'il ne s'agit pas d'une arborescence des appels de procédure ordonnée des appels d'API publics.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'action de CodeDeploy création d'un groupe de déploiement :

```
{
 "Records": [{
 "eventVersion": "1.02",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAI44QH8DHBEXAMPLE:203.0.113.11",
 "arn": "arn:aws:sts::123456789012:assumed-role/example-role/203.0.113.11",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2014-11-27T03:57:36Z"
 }
 },
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAI44QH8DHBEXAMPLE",
 "arn": "arn:aws:iam::123456789012:role/example-role",
 "accountId": "123456789012",
 "userName": "example-role"
 }
 },
 "eventTime": "2014-11-27T03:57:36Z",
 "eventSource": "codedeploy.amazonaws.com",
 "eventName": "CreateDeploymentGroup",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "203.0.113.11",
 "userAgent": "example-user-agent-string",
 "requestParameters": {
 "applicationName": "ExampleApplication",
 "serviceRoleArn": "arn:aws:iam::123456789012:role/example-instance-group-role",
 "deploymentGroupName": "ExampleDeploymentGroup",
 "ec2TagFilters": [{
 "value": "CodeDeployDemo",
 "type": "KEY_AND_VALUE",
 "key": "Name"
 }
]
 }
]
```



```
 }],
 "deploymentConfigName": "CodeDeployDefault.HalfAtATime"
 },
 "responseElements": {
 "deploymentGroupId": "7d64e680-e6f4-4c07-b10a-9e117EXAMPLE"
 },
 "requestID": "86168559-75e9-11e4-8cf8-75d18EXAMPLE",
 "eventID": "832b82d5-d474-44e8-a51d-093ccEXAMPLE",
 "eventType": "AwsApiCall",
 "recipientAccountId": "123456789012"
},
 ... additional entries ...
]
}
```

## Surveillance des déploiements à l'aide des notifications d'événements Amazon SNS

Vous pouvez ajouter des déclencheurs à un groupe de CodeDeploy déploiement pour recevoir des notifications concernant les événements liés aux déploiements ou aux instances de ce groupe de déploiement. Ces notifications sont envoyées aux destinataires abonnés à une rubrique Amazon SNS que vous avez intégrée à l'action du déclencheur.

Vous pouvez recevoir des notifications d' CodeDeploy événements par SMS ou par e-mail. Vous pouvez également utiliser les données JSON créées lorsqu'un événement spécifique se produit d'une autre manière, par exemple en envoyant des messages aux files d'attente Amazon SQS ou en invoquant une fonction dans AWS Lambda. Pour examiner la structure des données JSON fournies pour les déclencheurs de déploiement et d'instance, consultez la page [Formats de données JSON pour les CodeDeploy déclencheurs](#).

Vous pouvez choisir d'utiliser des déclencheurs pour recevoir des notifications si :

- vous êtes un développeur qui a besoin de savoir quand un déploiement échoue ou s'arrête afin de pouvoir le dépanner ;
- En tant qu'administrateur système, vous devez connaître le nombre d'instances défectueuses afin de surveiller l'état de santé de votre flotte Amazon EC2.
- Vous êtes un responsable qui souhaite connaître le at-a-glance nombre d'événements liés au déploiement et aux instances, ce que vous pouvez obtenir grâce à des règles de filtrage qui

acheminement différents types de notifications vers des dossiers de votre client de messagerie de bureau.

Vous pouvez créer jusqu'à 10 déclencheurs pour chaque groupe de CodeDeploy déploiement, pour l'un des types d'événements suivants.

| Événements de déploiement                                                                                                                                                                                          | Événements d'instance                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Réussite</li><li>• Échec</li><li>• Démarré(e)</li><li>• Arrêté(e)</li><li>• Restauration</li><li>• Prêt<sup>1</sup></li><li>• Tous les événements de déploiement</li></ul> | <ul style="list-style-type: none"><li>• Réussite</li><li>• Échec</li><li>• Démarré(e)</li><li>• Prêt<sup>1</sup></li><li>• Tous les événements d'instance</li></ul> |

<sup>1</sup> S'applique uniquement aux blue/green déploiements. Indique que la dernière révision de l'application a été installée sur les instances dans un environnement de remplacement et que le trafic à partir de l'environnement d'origine peut désormais être réacheminé derrière un équilibreur de charge. Pour de plus amples informations, veuillez consulter [Travailler avec des déploiements dans CodeDeploy](#).

## Rubriques

- [Accorder des autorisations Amazon SNS à un rôle de service CodeDeploy](#)
- [Création d'un déclencheur pour un CodeDeploy événement](#)
- [Modifier un déclencheur dans un groupe CodeDeploy de déploiement](#)
- [Supprimer un déclencheur d'un groupe CodeDeploy de déploiement](#)
- [Formats de données JSON pour les CodeDeploy déclencheurs](#)

## Accorder des autorisations Amazon SNS à un rôle de service CodeDeploy

Avant que vos déclencheurs puissent générer des notifications, le rôle de service que vous utilisez dans vos CodeDeploy opérations doit être autorisé à accéder aux ressources Amazon SNS.

## Pour accorder des autorisations Amazon SNS à un rôle de service

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le volet de navigation de la console IAM, sélectionnez Rôles.
3. Choisissez le nom du rôle de service que vous utilisez dans vos opérations AWS CodeDeploy .
4. Dans l'onglet Autorisations, dans la zone Stratégies en ligne, choisissez Créer une stratégie de rôle.

—ou—

Si le bouton Créer une stratégie de rôle n'est pas disponible, développez la zone Stratégies en ligne, puis choisissez cliquez ici.

5. Sur la page Set Permissions (Définir les autorisations), choisissez Stratégie personnalisée, puis Sélectionner.
6. Sur la page Examiner une stratégie, dans le champ Nom de la stratégie, entrez un nom pour identifier cette stratégie, comme SNSPublish.
7. Collez la stratégie suivante dans le champ Policy Document (Document de stratégie) :

JSON

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

8. Choisissez Apply Policy (Appliquer la stratégie).

## Création d'un déclencheur pour un CodeDeploy événement

Vous pouvez créer un déclencheur qui publie une rubrique Amazon Simple Notification Service (Amazon SNS) pour AWS CodeDeploy un événement de déploiement ou d'instance. Ensuite, lorsque

cet événement se produit, tous les abonnés à la rubrique associée reçoivent des notifications via le point de terminaison spécifié dans la rubrique, par exemple un SMS ou un e-mail. Amazon SNS propose plusieurs méthodes pour s'abonner à des rubriques.

Avant de créer un déclencheur, vous devez configurer la rubrique Amazon SNS vers laquelle le déclencheur doit pointer. Pour plus d'informations, voir [Création d'un sujet](#). Lorsque vous créez un sujet, nous vous recommandons de lui donner un nom identifiant son objectif, dans des formats tels que `Topic-group-us-west-3-deploy-fail` ou `Topic-group-project-2-instance-stop`.

Vous devez également accorder des autorisations Amazon SNS à un rôle de CodeDeploy service avant que des notifications puissent être envoyées pour votre déclencheur. Pour plus d'informations, consultez [Accorder des autorisations Amazon SNS à un rôle de service CodeDeploy](#).

Une fois que vous avez créé la rubrique, vous pouvez ajouter des abonnés. Pour plus d'informations sur la création, la gestion et l'abonnement à des rubriques, consultez [What is Amazon Simple Notification Service](#).

## Création d'un déclencheur pour envoyer des notifications CodeDeploy d'événements (console)

Vous pouvez utiliser la CodeDeploy console pour créer des déclencheurs pour un CodeDeploy événement. À la fin du processus d'installation, un message de notification test est envoyé pour garantir la configuration correcte des autorisations et des détails de déclencheur.

Pour créer un déclencheur pour un CodeDeploy événement

1. Dans le AWS Management Console, ouvrez la AWS CodeDeploy console.
2. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

### Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

3. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
4. Sur la page Applications, choisissez le nom de l'application associée au groupe de déploiement dans lequel vous allez ajouter un déclencheur.

5. Sur la page Application details (Détails de l'application), choisissez le groupe de déploiement dans lequel vous souhaitez ajouter un déclencheur.
6. Choisissez Modifier.
7. (Facultatif) Développez Advanced (Options avancées).
8. Dans la zone Triggers (Déclencheurs), choisissez Create trigger (Créer un déclencheur).
9. Dans le volet Create deployment trigger (Créer un déclencheur de déploiement) procédez comme suit :
  - a. Dans Nom du déclencheur, entrez pour le déclencheur un nom permettant d'identifier sa fonction. Nous vous recommandons des formats tels que `Trigger-group-us-west-3-deploy-fail` ou `Trigger-group-eu-central-instance-stop`.
  - b. Dans Events, choisissez le ou les types d'événements pour déclencher l'envoi de notifications par la rubrique Amazon SNS.
  - c. Dans les rubriques Amazon SNS, choisissez le nom de la rubrique que vous avez créée pour envoyer des notifications pour ce déclencheur.
  - d. Choisissez Créer un déclencheur. CodeDeploy envoie une notification de test pour confirmer que vous avez correctement configuré l'accès entre CodeDeploy et la rubrique Amazon SNS. Selon le type de point de terminaison que vous avez sélectionné pour la rubrique et le fait que vous vous soyez abonné ou non à la rubrique, vous recevez la confirmation dans un message SMS ou un e-mail.
10. Sélectionnez Enregistrer les modifications.

## Création d'un déclencheur pour envoyer des notifications d' CodeDeploy événements (CLI)

Vous pouvez utiliser l'interface CLI pour inclure des déclencheurs lorsque vous créez un groupe de déploiement, ou vous pouvez ajouter des déclencheurs à un groupe de déploiement existant.

Pour créer un déclencheur afin d'envoyer des notifications pour un nouveau groupe de déploiement

Créez un fichier JSON pour configurer le groupe de déploiement, puis exécutez la [create-deployment-group](#) commande à l'aide de l'`--cli-input-json` option.

La façon la plus simple de créer le fichier JSON consiste à utiliser l'option `--generate-cli-skeleton` pour obtenir une copie du format JSON, puis à fournir les valeurs requises dans un éditeur de texte brut.

1. Exécutez la commande suivante, puis copiez les résultats dans un éditeur de texte brut.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Ajoutez le nom d'une CodeDeploy application existante à la sortie :

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentGroupName": "",
 "deploymentConfigName": "",
 "ec2TagFilters": [
 {
 "Key": "",
 "Value": "",
 "Type": ""
 }
],
 "onPremisesInstanceTagFilters": [
 {
 "Key": "",
 "Value": "",
 "Type": ""
 }
],
 "autoScalingGroups": [
 ""
],
 "serviceRoleArn": "",
 "triggerConfigurations": [
 {
 "triggerName": "",
 "triggerTargetArn": "",
 "triggerEvents": [
 ""
]
 }
]
}
```

3. Fournissez des valeurs pour les paramètres que vous souhaitez configurer.

Lorsque vous utilisez la [create-deployment-group](#) commande, vous devez fournir, au minimum, des valeurs pour les paramètres suivants :

- `applicationName` : le nom d'une application déjà créée dans votre compte.
- `deploymentGroupName` : un nom pour le groupe de déploiement que vous créez.
- `serviceRoleArn`: ARN d'un rôle de service existant configuré pour CodeDeploy votre compte. Pour plus d'informations, consultez [Étape 2 : créer un rôle de service pour CodeDeploy](#).

Dans la section `triggerConfigurations`, fournissez des valeurs pour les paramètres suivants :

- `triggerName` : le nom que vous voulez donner au déclencheur pour pouvoir l'identifier facilement. Nous vous recommandons des formats tels que `Trigger-group-us-west-3-deploy-fail` ou `Trigger-group-eu-central-instance-stop`.
- `triggerTargetArn`: L'ARN de la rubrique Amazon SNS que vous avez créée pour l'associer à votre déclencheur, au format suivant : `arn:aws:sns:us-east-2:444455556666:NewTestTopic`
- `triggerEvents` : le type d'événement ou d'événements pour lequel vous souhaitez déclencher des notifications. Vous pouvez spécifier un ou plusieurs types d'événement, en séparant plusieurs noms de types d'événement par des virgules (par exemple, `"triggerEvents": ["DeploymentSuccess", "DeploymentFailure", "InstanceFailure"]`). Lorsque vous ajoutez plusieurs types d'événement, des notifications pour tous ces types sont envoyées à la rubrique que vous avez spécifiée, plutôt qu'à une rubrique différente pour chacun d'eux. Vous pouvez choisir parmi les types d'événement suivants :
  - `DeploymentStart`
  - `DeploymentSuccess`
  - `DeploymentFailure`
  - `DeploymentStop`
  - `DeploymentRollback`
  - `DeploymentReady` (S'applique uniquement aux instances de remplacement dans le cadre d'un blue/green déploiement)
  - `InstanceStart`
  - `InstanceSuccess`
  - `InstanceFailure`

- InstanceReady (S'applique uniquement aux instances de remplacement dans le cadre d'un blue/green déploiement)

L'exemple de configuration suivant crée un groupe de déploiement nommé `dep-group-ghi-789-2` pour une application nommée `TestApp-us-east-2` et un déclencheur qui vous invite à envoyer des notifications chaque fois qu'un déploiement commence, réussit ou échoue :

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "deploymentGroupName": "dep-group-ghi-789-2",
 "ec2TagFilters": [
 {
 "Key": "Name",
 "Value": "Project-ABC",
 "Type": "KEY_AND_VALUE"
 }
],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "triggerConfigurations": [
 {
 "triggerName": "Trigger-group-us-east-2",
 "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-
deployments",
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure"
]
 }
]
}
```

4. Enregistrez vos mises à jour sous la forme d'un fichier JSON, puis appelez ce fichier à l'aide de l'option `--cli-input-json` lorsque vous exécutez la commande `create-deployment-group` :

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.



```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

A la fin du processus de création, vous recevez un message de notification test qui indique que les autorisations et les détails de déclencheur sont correctement configurés.

Pour créer un déclencheur afin d'envoyer des notifications pour un groupe de déploiement existant

Pour utiliser le AWS CLI pour ajouter des déclencheurs d' CodeDeploy événements à un groupe de déploiement existant, créez un fichier JSON pour mettre à jour le groupe de déploiement, puis exécutez la [update-deployment-group](#) commande à l'aide de l'`--cli-input-json` option.

La façon la plus simple de créer le fichier JSON consiste à exécuter la commande `get-deployment-group` pour obtenir une copie de la configuration du groupe de déploiement, au format JSON, puis à mettre à jour les valeurs des paramètres dans un éditeur de texte brut.

1. Exécutez la commande suivante, puis copiez les résultats dans un éditeur de texte brut.

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```

2. Supprimez les éléments suivants de la sortie :

- Au début de la sortie, supprimez `{ "deploymentGroupInfo":`.
- A la fin de la sortie, supprimez `}`.
- Supprimez la ligne contenant `deploymentGroupId`.
- Supprimez la ligne contenant `deploymentGroupName`.

Le contenu de votre fichier texte doit être maintenant similaire à ce qui suit :

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "Project-ABC",
```

```
 "Key": "Name"
 }
],
 "triggerConfigurations": [],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "onPremisesInstanceTagFilters": []
 }
}
```

3. Dans la section `triggerConfigurations`, ajoutez des données pour les paramètres `triggerEvents`, `triggerTargetArn` et `triggerName`. Pour plus d'informations sur les paramètres de configuration des déclencheurs, consultez [TriggerConfig](#).

Le contenu de votre fichier texte doit être maintenant similaire à ce qui suit. Ce code invite à envoyer des notifications chaque fois qu'un déploiement commence, réussit ou échoue.

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "Project-ABC",
 "Key": "Name"
 }
],
 "triggerConfigurations": [
 {
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure"
],
 "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-deployments",
 "triggerName": "Trigger-group-us-east-2"
 }
],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "onPremisesInstanceTagFilters": []
}
```

4. Enregistrez vos mises à jour sous forme de fichier JSON, puis exécutez la [update-deployment-group](#) commande à l'aide de l'`--cli-input-json` option. Assurez-vous d'inclure l'`--current-deployment-group-name` option et de remplacer le nom de votre fichier JSON par *filename* :

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
```

À la fin du processus de création, vous recevez un message de notification test qui indique que les autorisations et les détails de déclencheur sont correctement configurés.

## Modifier un déclencheur dans un groupe CodeDeploy de déploiement

Si vos exigences de notification changent, vous pouvez modifier votre déclencheur plutôt qu'en créer un nouveau.

### Modifier un CodeDeploy déclencheur (CLI)

Pour utiliser le AWS CLI pour modifier les détails du déclencheur d' CodeDeploy événements lorsque vous mettez à jour un groupe de déploiement, créez un fichier JSON pour définir les modifications apportées aux propriétés du groupe de déploiement, puis exécutez la [update-deployment-group](#) commande avec l'`--cli-input-json` option.

La façon la plus simple de créer le fichier JSON consiste à exécuter la commande `get-deployment-group` pour obtenir les détails du groupe de déploiement actuel au format JSON, puis à modifier les valeurs requises dans un éditeur de texte brut.

1. Exécutez la commande suivante en remplaçant les noms de votre application et de votre groupe de déploiement par *application* et *deployment-group* :

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```

2. Copiez les résultats de la commande dans un éditeur de texte brut, puis supprimez les éléments suivants :

- Au début de la sortie, supprimez { "deploymentGroupInfo":.
- A la fin de la sortie, supprimez }.
- Supprimez la ligne contenant deploymentGroupId.
- Supprimez la ligne contenant deploymentGroupName.

Le contenu de votre fichier texte doit être maintenant similaire à ce qui suit :

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "East-1-Instances",
 "Key": "Name"
 }
],
 "triggerConfigurations": [
 {
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure",
 "DeploymentStop"
],
 "triggerTargetArn": "arn:aws:sns:us-east-2:111222333444:Trigger-group-us-east-2",
 "triggerName": "Trigger-group-us-east-2"
 }
],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "onPremisesInstanceTagFilters": []
}
```

3. Modifiez les paramètres nécessaires, le cas échéant. Pour plus d'informations sur les paramètres de configuration des déclencheurs, consultez [TriggerConfig](#).

4. Enregistrez vos mises à jour sous forme de fichier JSON, puis exécutez la [update-deployment-group](#) commande à l'aide de l'`--cli-input-json` option. Assurez-vous d'inclure l'`--current-deployment-group-name` option et de remplacer le nom de votre fichier JSON par *filename* :

 Important

N'oubliez pas d'inclure `file://` devant le nom du fichier. Il est nécessaire dans cette commande.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-group-name --cli-input-json file://filename.json
```

A la fin du processus de création, vous recevez un message de notification test qui indique que les autorisations et les détails de déclencheur sont correctement configurés.

## Supprimer un déclencheur d'un groupe CodeDeploy de déploiement

Comme il existe une limite de 10 déclencheurs par groupe de déploiement, vous pouvez supprimer des déclencheurs si vous ne les utilisez plus. Vous ne pouvez pas annuler la suppression d'un déclencheur, mais vous pouvez en recréer un.

### Supprimer un déclencheur d'un groupe de déploiement (console)

1. Connectez-vous à la CodeDeploy console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codedeploy>.

 Note

Connectez-vous avec le même utilisateur que celui que vous avez configuré [Commencer avec CodeDeploy](#).

2. Dans le volet de navigation, développez Deploy, puis choisissez Applications.
3. Sur la page Applications, choisissez le nom de l'application associée au groupe de déploiement dans lequel vous allez supprimer un déclencheur.
4. Sur la page Application details (Détails de l'application), choisissez le groupe de déploiement dans lequel vous souhaitez supprimer un déclencheur.

5. Choisissez Modifier.
6. (Facultatif) Développez Advanced (Options avancées).
7. Dans la zone Triggers (Déclencheurs), choisissez le déclencheur à supprimer, puis Delete trigger (Supprimer le déclencheur).
8. Sélectionnez Enregistrer les modifications.

## Supprimer un déclencheur d'un groupe de déploiement (CLI)

Pour utiliser l'interface CLI afin de supprimer un déclencheur, appelez la commande [update-deployment-group](#), avec des paramètres de configuration de déclencheur vides, en spécifiant :

- Le nom de l'application associée au groupe de déploiement. Pour afficher la liste des noms d'applications, appelez la commande [list-applications](#).
- Le nom du groupe de déploiement associé à l'application. Pour afficher la liste des noms des groupes de déploiement, appelez la commande [list-deployment-groups](#).

Par exemple :

```
aws deploy update-deployment-group --application-name application-name --current-deployment-group-name deployment-group-name --trigger-configurations
```

## Formats de données JSON pour les CodeDeploy déclencheurs

Vous pouvez utiliser la sortie JSON créée lorsqu'un déclencheur pour un déploiement ou une instance est activé dans un flux de notification personnalisé, tel que l'envoi de messages aux files d'attente Amazon SQS ou l'appel d'une fonction dans. AWS Lambda

### Note

Ce guide ne traite pas de la façon de configurer les notifications à l'aide de JSON. Pour plus d'informations sur l'utilisation d'Amazon SNS pour envoyer des messages aux files d'attente Amazon SQS, consultez la section Envoi de [messages Amazon SNS](#) aux files d'attente Amazon SQS. Pour plus d'informations sur l'utilisation d'Amazon SNS pour appeler une fonction Lambda, consultez Invoquer des [fonctions Lambda à l'aide des notifications](#) Amazon SNS.

Les exemples suivants illustrent la structure de la sortie JSON disponible avec les déclencheurs CodeDeploy.

### Exemple de sortie JSON pour les déclencheurs basés sur une instance

```
{
 "region": "us-east-2",
 "accountId": "111222333444",
 "eventTriggerName": "trigger-group-us-east-instance-succeeded",
 "deploymentId": "d-75I7MBT7C",
 "instanceId": "arn:aws:ec2:us-east-2:444455556666:instance/i-496589f7",
 "lastUpdatedAt": "1446744207.564",
 "instanceStatus": "Succeeded",
 "lifecycleEvents": [
 {
 "LifecycleEvent": "ApplicationStop",
 "LifecycleEventStatus": "Succeeded",
 "StartTime": "1446744188.595",
 "EndTime": "1446744188.711"
 },
 {
 "LifecycleEvent": "BeforeInstall",
 "LifecycleEventStatus": "Succeeded",
 "StartTime": "1446744189.827",
 "EndTime": "1446744190.402"
 }
]
 //More lifecycle events might be listed here
}
```

### Exemple de sortie JSON pour les déclencheurs basés sur un déploiement

```
{
 "region": "us-west-1",
 "accountId": "111222333444",
 "eventTriggerName": "Trigger-group-us-west-3-deploy-failed",
 "applicationName": "ProductionApp-us-west-3",
 "deploymentId": "d-75I7MBT7C",
 "deploymentGroupName": "dep-group-def-456",
 "createTime": "1446744188.595",
 "completeTime": "1446744190.402",
 "deploymentOverview": {
 "Failed": "10",
```

```
 "InProgress": "0",
 "Pending": "0",
 "Skipped": "0",
 "Succeeded": "0"
 },
 "status": "Failed",
 "errorInformation": {
 "ErrorCode": "IAM_ROLE_MISSING",
 "ErrorMessage": "IAM Role is missing for deployment group: dep-group-def-456"
 }
}
```



# Sécurité dans AWS CodeDeploy

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture 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 CodeDeploy, 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, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation CodeDeploy. Les rubriques suivantes expliquent comment procéder à la configuration CodeDeploy 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 CodeDeploy ressources.

## Rubriques

- [Protection des données dans AWS CodeDeploy](#)
- [Gestion des identités et des accès pour AWS CodeDeploy](#)
- [Connexion et surveillance CodeDeploy](#)
- [Validation de conformité pour AWS CodeDeploy](#)
- [Résilience dans AWS CodeDeploy](#)
- [Sécurité de l'infrastructure dans AWS CodeDeploy](#)

# Protection des données dans AWS CodeDeploy

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS CodeDeploy. 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 CodeDeploy 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.

## Confidentialité du trafic inter-réseau

CodeDeploy est un service de déploiement entièrement géré qui prend en charge les instances EC2, les fonctions Lambda, Amazon ECS et les serveurs sur site. Pour les instances EC2 et les serveurs sur site, un agent basé sur l'hôte communique à l'aide du protocole TLS. CodeDeploy

Actuellement, la communication entre l'agent et le service nécessite une connexion Internet sortante afin que l'agent puisse communiquer avec le public CodeDeploy et les points de terminaison du service Amazon S3. Dans un VPC (virtual private cloud), cela peut être réalisé avec une passerelle Internet, une connexion VPN de site à site à votre réseau d'entreprise ou une connexion directe.

L' CodeDeploy agent prend en charge les proxys HTTP.

Les points de terminaison Amazon VPC, alimentés par AWS PrivateLink, sont disponibles CodeDeploy dans certaines régions. Pour en savoir plus, consultez [Utilisation CodeDeploy avec Amazon Virtual Private Cloud](#).

### Note

L' CodeDeploy agent n'est requis que si vous effectuez un déploiement sur une plateforme informatique Amazon EC2/on-premises. L'agent n'est pas requis pour les déploiements utilisant Amazon ECS ou une plate-forme de AWS Lambda calcul.

## Chiffrement au repos

Le code client n'est pas stocké dans CodeDeploy. En tant que service de déploiement, CodeDeploy envoie des commandes à l' CodeDeploy agent exécuté sur des instances EC2 ou des serveurs sur site. L' CodeDeploy agent exécute ensuite les commandes à l'aide du protocole TLS. Les données du modèle de service pour les déploiements, la configuration des déploiements, les groupes de déploiement, les applications et les révisions d'applications sont stockées dans Amazon DynamoDB et chiffrées au repos à Clé détenue par AWS l'aide d'un, détenu et géré par. CodeDeploy Pour plus d'informations, consultez [Clé détenue par AWS s.](#)

## Chiffrement en transit

L'agent CodeDeploy initie toutes les communications CodeDeploy via le port 443. L'agent interroge CodeDeploy et écoute une commande. L'agent CodeDeploy est open source. Toutes les communications client-to-service sont cryptées en transit à l'aide du protocole TLS. Cela protège les données des clients en transit entre CodeDeploy et d'autres services tels qu'Amazon S3.

## Gestion des clés de chiffrement

Il n'y a aucune clé de chiffrement à gérer. Les données du modèle de CodeDeploy service sont cryptées à l'aide d'une clé détenue par AWS, détenues et gérées par CodeDeploy. Pour plus d'informations, consultez [Clé détenue par AWS s.](#)

## Gestion des identités et des accès pour AWS CodeDeploy

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser CodeDeploy les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

### Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Comment AWS CodeDeploy fonctionne avec IAM](#)
- [AWS politiques gérées \(prédéfinies\) pour CodeDeploy](#)
- [CodeDeploy mises à jour des politiques AWS gérées](#)
- [AWS CodeDeploy exemples de politiques basées sur l'identité](#)
- [Résolution des problèmes AWS CodeDeploy d'identité et d'accès](#)
- [CodeDeploy référence aux autorisations](#)
- [Prévention du cas de figure de l'adjoint désorienté entre services](#)

## Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résolution des problèmes AWS CodeDeploy d'identité et d'accès](#))
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment AWS CodeDeploy fonctionne avec IAM](#))
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir [AWS CodeDeploy exemples de politiques basées sur l'identité](#))

## Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [Signature AWS Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

### Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez [Tâches qui requièrent les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

### Utilisateurs et groupes

Un [utilisateur IAM](#) est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le guide de l'utilisateur IAM.

[Les groupes IAM](#) spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez [Cas d'utilisation pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un [rôle IAM](#) est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en [passant d'un rôle d'utilisateur à un rôle IAM \(console\)](#) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

## Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

## Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de l'utilisateur AWS Organizations.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

## Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

## Comment AWS CodeDeploy fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à CodeDeploy, vous devez comprendre quelles fonctionnalités IAM sont disponibles avec. CodeDeploy Pour plus d'informations, consultez la section [AWS Services compatibles avec IAM](#) dans le Guide de l'utilisateur d'IAM.

## Rubriques



- [CodeDeploy politiques basées sur l'identité](#)
- [CodeDeploy politiques basées sur les ressources](#)
- [Autorisation basée sur les CodeDeploy tags](#)
- [CodeDeploy Rôles IAM](#)

## CodeDeploy politiques basées sur l'identité

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. CodeDeploy prend en charge les actions, les ressources et les clés de condition. Pour plus d'informations sur les éléments que vous utilisez dans une stratégie JSON, consultez la [référence des éléments de stratégie JSON IAM](#) dans le guide de l'utilisateur IAM.

### Actions

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions de politique en CodeDeploy cours utilisent le `codedeploy:` préfixe avant l'action. Par exemple, l'autorisation `codedeploy:GetApplication` accorde à l'utilisateur l'autorisation d'effectuer l'opération `GetApplication`. Les déclarations de politique doivent inclure un `NotAction` élément `Action` ou. CodeDeploy définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": [
 "codedeploy:action1",
 "codedeploy:action2"
```

Vous pouvez aussi préciser plusieurs actions à l'aide de caractères génériques (\*). Par exemple, incluez l'action suivante pour spécifier toutes les actions qui commencent par le mot `Describe` :



```
"Action": "ec2:Describe"
```

Pour obtenir la liste des CodeDeploy actions, reportez-vous à la section [Actions définies par AWS CodeDeploy](#) dans le guide de l'utilisateur IAM.

Pour un tableau répertoriant toutes les actions d' CodeDeploy API et les ressources auxquelles elles s'appliquent, consultez [CodeDeploy référence aux autorisations](#).

## Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (\*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": ""
```

Par exemple, vous pouvez indiquer un groupe de déploiement (*myDeploymentGroup*) dans votre instruction à l'aide de son ARN comme suit :

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

Vous pouvez également spécifier tous les groupes de déploiement appartenant à un compte en utilisant le caractère générique (\*) comme suit :

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Pour spécifier toutes les ressources, ou si une action d'API n'est pas prise en charge ARNs, utilisez le caractère générique (\*) dans l'élément `Resource` comme suit :

```
"Resource": ""
```

Certaines actions CodeDeploy d'API acceptent plusieurs ressources (par exemple, `BatchGetDeploymentGroups`). Pour spécifier plusieurs ressources dans une seule instruction, séparez-les ARNs par des virgules, comme suit :

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy fournit un ensemble d'opérations permettant de travailler avec les CodeDeploy ressources. Pour obtenir la liste des opérations disponibles, consultez [CodeDeploy référence aux autorisations](#).

Pour obtenir une liste des types de CodeDeploy ressources et de leurs caractéristiques ARNs, consultez la section [Ressources définies par AWS CodeDeploy](#) dans le guide de l'utilisateur IAM. Pour plus d'informations sur les actions dans lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez la section [Actions définies par AWS CodeDeploy](#).

## CodeDeploy ressources et opérations

Dans CodeDeploy, la ressource principale est un groupe de déploiement. Dans une politique, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la politique s'applique. CodeDeploy prend en charge d'autres ressources qui peuvent être utilisées avec les groupes de déploiement, notamment les applications, les configurations de déploiement et les instances. Celles-ci sont appelées sous-ressources. Ces ressources et sous-ressources sont ARNs associées à des éléments uniques. Pour plus d'informations, consultez les [noms des ressources Amazon \(ARNs\)](#) dans le Référence générale d'Amazon Web Services.

| Type de ressource            | Format ARN                                                                                                    |
|------------------------------|---------------------------------------------------------------------------------------------------------------|
| Groupe de déploiement        | arn:aws:codedeploy: <i>region:account-id</i> :deploymentgroup: <i>application-name /deployment-group-name</i> |
| Application                  | arn:aws:codedeploy: <i>region:account-id</i> :application: <i>application-name</i>                            |
| Configuration de déploiement | arn:aws:codedeploy: <i>region:account-id</i> :deploymentconfig: <i>deployment-configuration-name</i>          |
| Instance                     | arn:aws:codedeploy: <i>region:account-id</i> :instance / <i>instance-ID</i>                                   |

| Type de ressource                                                                         | Format ARN                                                          |
|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| Toutes les CodeDeploy ressources                                                          | <code>arn:aws:codedeploy:*</code>                                   |
| Toutes les CodeDeploy ressources détenues par le compte spécifié dans la région spécifiée | <code>arn:aws:codedeploy: <i>region</i>:<i>account-id</i> :*</code> |

#### Note

La plupart des services AWS traitent deux points (:) ou une barre oblique (/) comme le même caractère dans ARNs. Cependant, CodeDeploy utilise une correspondance exacte dans les modèles de ressources et les règles. Veuillez à utiliser les caractères ARN corrects lors de la création de modèles d'événements, afin qu'ils correspondent à la syntaxe ARN de la ressource.

## Clés de condition

CodeDeploy ne fournit aucune clé de condition spécifique au service, mais il prend en charge l'utilisation de certaines clés de condition globales. Pour plus d'informations, consultez [Clés de contexte de condition globales AWS](#) dans le Guide de l'utilisateur IAM.

## Exemples

Pour consulter des exemples de politiques CodeDeploy basées sur l'identité, consultez. [AWS CodeDeploy exemples de politiques basées sur l'identité](#)

## CodeDeploy politiques basées sur les ressources

CodeDeploy ne prend pas en charge les politiques basées sur les ressources. Pour consulter un exemple de page de stratégie basée sur les ressources détaillée, consultez la section [Utilisation de politiques basées sur les ressources](#) pour. AWS Lambda

## Autorisation basée sur les CodeDeploy tags

CodeDeploy ne prend pas en charge le balisage des ressources ni le contrôle de l'accès en fonction des balises.

## CodeDeploy Rôles IAM

Un [rôle IAM](#) est une entité de votre AWS compte dotée d'autorisations spécifiques.

### Utilisation d'informations d'identification temporaires avec CodeDeploy

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à l'aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que [AssumeRole](#) ou [GetFederationToken](#).

CodeDeploy prend en charge l'utilisation d'informations d'identification temporaires.

### Rôles liés à un service

CodeDeploy ne prend pas en charge les rôles liés à un service.

### Rôles du service

Cette fonction permet à un service d'endosser une [fonction du service](#) en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les rôles de service apparaissent dans votre AWS compte et sont détenus par le compte. Cela signifie qu'un utilisateur peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

CodeDeploy prend en charge les rôles de service.

### Choisir un rôle IAM dans CodeDeploy

Lorsque vous créez une ressource de groupe de déploiement dans CodeDeploy, vous devez choisir un rôle pour autoriser CodeDeploy l'accès à Amazon EC2 en votre nom. Si vous avez déjà créé un rôle de service ou un rôle lié à un service, CodeDeploy vous fournit une liste de rôles dans laquelle effectuer votre choix. Il est important de choisir un rôle qui permet d'accéder au démarrage et à l'arrêt des instances EC2.

# AWS politiques gérées (prédéfinies) pour CodeDeploy

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 des autorisations pour les cas d'utilisation courants afin que vous n'ayez pas à rechercher les autorisations requises. Pour plus d'informations, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

## Rubriques

- [Liste des politiques AWS gérées pour CodeDeploy](#)
- [CodeDeploy politiques et notifications gérées](#)

## Liste des politiques AWS gérées pour CodeDeploy

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à CodeDeploy :

- `AWSCodeDeployFullAccess` : accorde un accès complet à CodeDeploy.

### Note

`AWSCodeDeployFullAccess` ne fournit pas d'autorisations pour les opérations dans d'autres services nécessaires au déploiement de vos applications, tels qu'Amazon EC2 et Amazon S3, uniquement pour les opérations spécifiques à CodeDeploy.

- `AWSCodeDeployDeployerAccess` : accorde l'autorisation d'enregistrer et de déployer des révisions.
- `AWSCodeDeployReadOnlyAccess` : accorde l'accès en lecture seule à CodeDeploy.
- `AWSCodeDeployRole` : Permet CodeDeploy de :
  - lire les balises de vos instances ou identifiez vos instances Amazon EC2 par les noms de groupes Amazon EC2 Auto Scaling
  - lire, créer, mettre à jour et supprimer des groupes Amazon EC2 Auto Scaling, des hooks de cycle de vie, des politiques de dimensionnement et des fonctionnalités de warm pool

- publier des informations dans des rubriques Amazon SNS
- récupérer des informations sur les CloudWatch alarmes Amazon
- lire et mettre à jour les ressources du service Elastic Load Balancing

La politique contient le code suivant :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "autoscaling:CompleteLifecycleAction",
 "autoscaling>DeleteLifecycleHook",
 "autoscaling:DescribeAutoScalingGroups",
 "autoscaling:DescribeLifecycleHooks",
 "autoscaling:PutLifecycleHook",
 "autoscaling:RecordLifecycleActionHeartbeat",
 "autoscaling>CreateAutoScalingGroup",
 "autoscaling>CreateOrUpdateTags",
 "autoscaling:UpdateAutoScalingGroup",
 "autoscaling:EnableMetricsCollection",
 "autoscaling:DescribePolicies",
 "autoscaling:DescribeScheduledActions",
 "autoscaling:DescribeNotificationConfigurations",
 "autoscaling:SuspendProcesses",
 "autoscaling:ResumeProcesses",
 "autoscaling:AttachLoadBalancers",
 "autoscaling:AttachLoadBalancerTargetGroups",
 "autoscaling:PutScalingPolicy",
 "autoscaling:PutScheduledUpdateGroupAction",
 "autoscaling:PutNotificationConfiguration",
 "autoscaling:DescribeScalingActivities",
 "autoscaling>DeleteAutoScalingGroup",
 "autoscaling:PutWarmPool",
 "ec2:DescribeInstances",
 "ec2:DescribeInstanceStatus",
 "ec2:TerminateInstances",
 "tag:GetResources",
 "sns:Publish",
 "cloudwatch:DescribeAlarms",
```

```

 "cloudwatch:PutMetricAlarm",
 "elasticloadbalancing:DescribeLoadBalancers",
 "elasticloadbalancing:DescribeLoadBalancerAttributes",
 "elasticloadbalancing:DescribeInstanceHealth",
 "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
 "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeTargetGroupAttributes",
 "elasticloadbalancing:DescribeTargetHealth",
 "elasticloadbalancing:RegisterTargets",
 "elasticloadbalancing:DeregisterTargets"
],
 "Resource": "*"
}
]
}

```

- **AWSCodeDeployRoleForLambda**: accorde CodeDeploy l'autorisation d'accès AWS Lambda et toute autre ressource requise pour un déploiement.
- **AWSCodeDeployRoleForECS**: accorde CodeDeploy l'autorisation d'accéder à Amazon ECS et à toute autre ressource requise pour un déploiement.
- **AWSCodeDeployRoleForECSLimited**: accorde CodeDeploy l'autorisation d'accéder à Amazon ECS et à toute autre ressource requise pour un déploiement, avec les exceptions suivantes :
  - Dans la `hooks` section du AppSpec fichier, seules les fonctions Lambda dont le nom commence par `CodeDeployHook_` peuvent être utilisées. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement Amazon ECS](#).
  - L'accès aux compartiments S3 est limité aux compartiments S3 dotés d'une balise d'enregistrement, `UseWithCodeDeploy`, qui a la valeur `true`. Pour plus d'informations, consultez la section [Balisage d'objets](#).
- **AmazonEC2RoleforAWSCodeDeployLimited**: accorde CodeDeploy l'autorisation d'obtenir et de répertorier des objets dans un compartiment CodeDeploy Amazon S3. La politique contient le code suivant :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion",
 "s3:ListBucket"
],
 "Resource": "arn:aws:s3:::*/CodeDeploy/*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
 }
 }
 }
]
}
```

Les autorisations relatives à certains aspects du processus de déploiement sont accordées à deux autres types de rôles qui agissent pour le compte de CodeDeploy :

- Un profil d'instance IAM est un rôle IAM que vous attachez à vos instances Amazon EC2. Ce profil inclut les autorisations requises pour accéder aux compartiments ou aux GitHub référentiels Amazon S3 dans lesquels les applications sont stockées. Pour de plus amples informations, veuillez consulter [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).
- Un rôle de service est un rôle IAM qui accorde des autorisations à un AWS service afin qu'il puisse accéder aux AWS ressources. Les politiques que vous attachez au rôle de service déterminent



les AWS ressources auxquelles le service peut accéder et les actions qu'il peut effectuer avec ces ressources. En CodeDeploy effet, un rôle de service est utilisé pour les opérations suivantes :

- Pour lire les balises appliquées aux instances ou les noms de groupes Amazon EC2 Auto Scaling associés aux instances. Cela permet d' CodeDeploy identifier les instances sur lesquelles il peut déployer des applications.
- Pour effectuer des opérations sur des instances, des groupes Amazon EC2 Auto Scaling et des équilibreurs de charge Elastic Load Balancing.
- Publier des informations dans les rubriques Amazon SNS afin que des notifications puissent être envoyées lorsque des événements de déploiement ou d'instance spécifiés se produisent.
- Pour récupérer des informations sur les CloudWatch alarmes afin de configurer la surveillance des alarmes pour les déploiements.

Pour de plus amples informations, veuillez consulter [Étape 2 : créer un rôle de service pour CodeDeploy](#).

Vous pouvez également créer des politiques IAM personnalisées pour accorder des autorisations pour les CodeDeploy actions et les ressources. Vous associez ces politiques personnalisées aux rôles IAM, puis vous attribuez les rôles aux utilisateurs ou aux groupes qui ont besoin des autorisations.

## CodeDeploy politiques et notifications gérées

CodeDeploy prend en charge les notifications pour informer les utilisateurs des modifications importantes apportées aux déploiements. Les politiques gérées CodeDeploy incluent des déclarations de politique relatives à la fonctionnalité de notification. Pour plus d'informations, consultez [En quoi consistent les notifications ?](#)

### Autorisations pour les notifications dans les politiques gérées à accès complet

La stratégie gérée `AWSCodeDeployFullAccess` inclut les déclarations suivantes pour permettre un accès complet aux notifications. Les utilisateurs auxquels cette politique gérée est appliquée peuvent également créer et gérer des sujets Amazon SNS pour les notifications, abonner et désinscrire des utilisateurs à des sujets, et répertorier les sujets à choisir comme cibles pour les règles de notification.

```
{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
```

```

 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications>DeleteNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition" : {
 "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codedeploy:*:*:application:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource"
],
 "Resource": "*"
},
{
 "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
 "Effect": "Allow",
 "Action": [
 "sns:CreateTopic",
 "sns:SetTopicAttributes"
],
 "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
 "Sid" : "CodeStarNotificationsChatbotAccess",
 "Effect" : "Allow",
 "Action" : [
 "chatbot:DescribeSlackChannelConfigurations"
],
 "Resource" : "*"
},
{
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",

```

```
"Action": [
 "sns:ListTopics"
],
"Resource": "*"
}
```

## Autorisations pour les notifications dans les politiques gérées en lecture seule

La stratégie gérée `AWSCodeDeployReadOnlyAccess` 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:codedeploy:*:*:application:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules"
],
 "Resource": "*"
}
```

## Autorisations pour les notifications dans d'autres politiques gérées

La politique `AWSCodeDeployDeployerAccess` gérée inclut les instructions suivantes pour permettre aux utilisateurs de créer, de mettre à jour, de s'abonner et de consulter les notifications relatives aux ressources, mais ne peuvent pas les supprimer. Les utilisateurs auxquels cette politique gérée est appliquée peuvent également créer et gérer des rubriques Amazon SNS pour les notifications.

Cette politique inclut les autorisations pour effectuer les opérations suivantes :

- `codestar-notifications:CreateNotificationRule`— Permet aux directeurs de créer des notifications.
- `codestar-notifications:DescribeNotificationRule`— Permet aux principaux de récupérer des informations sur les notifications.
- `codestar-notifications:UpdateNotificationRule`— Permet aux principaux de mettre à jour les notifications.
- `codestar-notifications:Subscribe`— Permet aux directeurs de s'abonner aux mises à jour des notifications.
- `codestar-notifications:Unsubscribe`— Permet aux directeurs de se désabonner des mises à jour des notifications.
- `codestar-notifications:ListNotificationRules`— Permet aux principaux de récupérer la liste des règles de notification.
- `codestar-notifications:ListTargets`— Permet aux principaux de récupérer la liste des cibles.
- `codestar-notifications:ListTagsForResource`— Permet aux principaux de récupérer la liste des tags.
- `codestar-notifications:ListEventTypes`— Permet aux principaux de récupérer la liste des types d'événements.
- `chatbot:DescribeSlackChannelConfiguration`— Permet aux directeurs de récupérer des informations sur les configurations des chaînes Slack.
- `sns:ListTopics`— Permet aux principaux de récupérer la liste des sujets Amazon SNS pour les notifications.

```
{
 "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:codedeploy:*:*:application:*"
 }
 },
 },
 {
 "Sid" : "CodeStarNotificationsListAccess",
 "Effect" : "Allow",
 "Action" : [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource",
 "codestar-notifications:ListEventTypes"
],
 "Resource" : "*"
 },
 {
 "Sid" : "CodeStarNotificationsChatbotAccess",
 "Effect" : "Allow",
 "Action" : [
 "chatbot:DescribeSlackChannelConfigurations"
],
 "Resource" : "*"
 },
 {
 "Sid" : "SNSTopicListAccess",
 "Effect" : "Allow",
 "Action" : [
 "sns:ListTopics"
],
 "Resource" : "*"
 }
}

```

Pour plus d'informations, consultez la section [Gestion des identités et des accès pour CodeStar les notifications AWS](#).

## CodeDeploy mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CodeDeploy depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les

modifications apportées à cette page, abonnez-vous au flux RSS de la page CodeDeploy [Historique du document](#).

| Modifier                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Date             |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AWSCodeDeployDeployerAccess — Mises à jour de la politique existante | <p>L'codestar-notifications:NotificationsForResource action a été mise à jour pour prendre en charge les modifications de validation des politiques IAM. La ressource d'origine <code>arn:aws:codedeploy:*:*:application:*</code> a été mise à jour vers <code>arn:aws:codedeploy:*:*:application:*</code>.</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">Autorisations pour les notifications dans d'autres politiques gérées</a>.</p> | 16 décembre 2024 |
| AWSCodeDeployReadOnlyAccess — Mises à jour de la politique existante | <p>L'codestar-notifications:NotificationsForResource action a été mise à jour pour prendre en charge les modifications de validation des politiques IAM. La ressource d'origine <code>arn:aws:codedeploy:*:*:application:*</code> a été mise à jour vers <code>arn:aws:codedeploy:*:*:application:*</code>.</p> <p>Pour plus d'informations sur cette politique, consultez</p>                                                                                       | 16 décembre 2024 |

| Modifier                                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                         | Date             |
|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
|                                                                                  | <a href="#">Autorisations pour les notifications dans les politiques gérées en lecture seule.</a>                                                                                                                                                                                                                                                                                                                                   |                  |
| AWSCodeDeployFullAccess stratégie gérée — Mises à jour de la politique existante | <p>L'codestar-notifications:NotificationsForResource action a été mise à jour pour prendre en charge les modifications de validation des politiques IAM. La ressource d'origine arn:aws:codedeploy:* a été mise à jour versarn:aws:codedeploy:*:*:application:*</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">Autorisations pour les notifications dans les politiques gérées à accès complet.</a></p> | 16 décembre 2024 |

| Modifier                                                                         | Description                                                                                                                                                                                                                                                                                                                                                               | Date         |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| AWSCodeDeployRole stratégie gérée — Mises à jour de la politique existante       | <p>Ajout des <code>elasticloadbalancing:DescribeTargetGroupAttributes</code> actions <code>elasticloadbalancing:DescribeLoadBalancerAttributes</code> et à la déclaration de politique pour prendre en charge les modifications apportées à Elastic Load Balancing.</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">AWSCodeDeployRole</a>.</p> | 16 août 2023 |
| AWSCodeDeployFullAccess stratégie gérée — Mises à jour de la politique existante | <p>L'action <code>ListMicrosoftTeamsChannelConfigurations</code> a été ajoutée à la déclaration de politique pour prendre en charge les modifications apportées aux notifications.</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">AWSCodeDeployRole</a>.</p>                                                                                  | 11 mai 2023  |



| Modifier                                                                                      | Description                                                                                                                                                                                                                                                                                   | Date             |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AWSCodeDeployRole stratégie gérée — Mises à jour de la politique existante                    | <p>L' <code>autoscaling:CreateOrUpdateTags</code> action a été ajoutée à la déclaration de politique pour prendre en charge les modifications d'autorisation d'Amazon EC2 Auto Scaling.</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">AWSCodeDeployRole</a>.</p> | 3 février 2023   |
| AmazonEC2RoleforAWSCodeDeployLimited stratégie gérée — Mises à jour de la politique existante | <p>L'<code>s3:ListBucket</code> action a été supprimée de la déclaration de politique qui inclut la <code>s3:ExistingObjectTag/UseWithCodeDeploy</code> condition.</p> <p>Pour plus d'informations sur cette politique, consultez <a href="#">AmazonEC2RoleforAWSCodeDeployLimited</a>.</p>   | 22 novembre 2021 |
| AWSCodeDeployRole stratégie gérée — Mises à jour de la politique existante                    | <p>Ajout de l'<code>autoscaling:PutWarmPool</code> action permettant d'<a href="#">ajouter des pools de chaleur aux groupes Amazon EC2 Auto Scaling</a> pour les déploiements bleu/vert.</p> <p>Suppression des actions dupliquées inutiles.</p>                                              | 18 mai 2021      |

| Modifier                                         | Description                                                                             | Date        |
|--------------------------------------------------|-----------------------------------------------------------------------------------------|-------------|
| CodeDeploy a commencé à suivre les modifications | CodeDeploy a commencé à suivre les modifications apportées AWS à ses politiques gérées. | 18 mai 2021 |

## AWS CodeDeploy exemples de politiques basées sur l'identité

Par défaut, les utilisateurs ne sont pas autorisés à créer ou à modifier CodeDeploy des ressources. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Vous devez créer des politiques IAM qui accordent aux rôles IAM l'autorisation d'effectuer des opérations d'API sur les ressources spécifiées dont ils ont besoin. Vous devez ensuite associer ces rôles IAM aux utilisateurs ou aux groupes qui ont besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques dans l'onglet JSON](#) dans le Guide de l'utilisateur IAM.

Dans CodeDeploy, les politiques basées sur l'identité sont utilisées pour gérer les autorisations d'accès aux différentes ressources liées au processus de déploiement. Vous pouvez contrôler l'accès aux types de ressources suivants :

- Applications et révisions d'application
- Déploiements
- Configurations de déploiement
- Instances et instances sur site

Les capacités contrôlées par des stratégies de ressource varient en fonction du type de ressource, comme décrit dans le tableau suivant :

| Types de ressources | Fonctionnalités                                                   |
|---------------------|-------------------------------------------------------------------|
| Tous                | Afficher et répertorier les informations relatives aux ressources |
| Applications        | Créer des ressources                                              |

| Types de ressources           | Fonctionnalités                         |
|-------------------------------|-----------------------------------------|
| Configurations de déploiement | Supprimer des ressources                |
| Groupes de déploiement        |                                         |
| Déploiements                  | Créer des déploiements                  |
|                               | Arrêter des déploiements                |
| Révisions d'application       | Enregistrer les révisions d'application |
| Applications                  | Mettre à jour les ressources            |
| Groupes de déploiement        |                                         |
| Instances sur site            | Ajouter des balises aux instances       |
|                               | Supprimer des balises des instances     |
|                               | Enregistrer des instances               |
|                               | Annuler l'enregistrement des instances  |

L'exemple suivant présente une stratégie d'autorisation qui permet à un utilisateur de supprimer le groupe de déploiement nommé **WordPress\_DepGroup** associé à l'application nommée **WordPress\_App** dans la région **us-west-2**.

JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:DeleteDeploymentGroup"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/WordPress_DepGroup"
]
 }
]
}
```

```
]
}
]
}
```

## Rubriques

- [Exemples de politiques gérées par le client](#)
- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la CodeDeploy console](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

## Exemples de politiques gérées par le client

Dans cette section, vous trouverez des exemples de politiques qui accordent des autorisations pour diverses CodeDeploy actions. Ces politiques fonctionnent lorsque vous utilisez l' API CodeDeploy AWS SDKs, ou le AWS CLI. Vous devez accorder des autorisations supplémentaires pour les actions que vous effectuez dans la console. Pour en savoir plus sur l'octroi d'autorisations de console, consultez [Utilisation de la CodeDeploy console](#).

### Note

Tous les exemples utilisent la région de l'Ouest des États-Unis (Oregon) (`us-west-2`) et contiennent un récit fictif. IDs

## Exemples

- [Exemple 1 : Autoriser l'autorisation d'effectuer CodeDeploy des opérations dans une seule région](#)
- [Exemple 2 : Autoriser l'enregistrement des révisions pour une seule application](#)
- [Exemple 3 : Autoriser la création de déploiements pour un seul groupe de déploiement](#)

Exemple 1 : Autoriser l'autorisation d'effectuer CodeDeploy des opérations dans une seule région

L'exemple suivant accorde des autorisations pour effectuer CodeDeploy des opérations dans la **us-west-2** région uniquement :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:*"
]
 }
]
}
```

Exemple 2 : Autoriser l'enregistrement des révisions pour une seule application

L'exemple suivant accorde des autorisations pour enregistrer les révisions de toutes les applications qui commencent par **Test** dans la région **us-west-2** :

## JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:RegisterApplicationRevision"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:application:Test*"
]
 }
]
}
```

### Exemple 3 : Autoriser la création de déploiements pour un seul groupe de déploiement

L'exemple suivant autorise la création de déploiements pour le groupe de déploiement nommé **WordPress\_DepGroup** associé à l'application nommée **WordPress\_App**, la configuration de déploiement personnalisée nommée **ThreeQuartersHealthy** et toutes les révisions d'application associées à l'application nommée **WordPress\_App**. Toutes ces ressources se trouvent dans la région **us-west-2**.

JSON

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:CreateDeployment"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:deploymentgroup:WordPress_App/WordPress_DepGroup"
]
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:GetDeploymentConfig"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:deploymentconfig:ThreeQuartersHealthy"
]
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:GetApplicationRevision"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:111122223333:application:WordPress_App"
]
 }
]
}
```

```
]
}
```

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer CodeDeploy des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles.

Pour plus d'informations, consultez [Validation de politiques avec IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Sécurisation de l'accès aux API avec MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

## Utilisation de la CodeDeploy console

Si vous utilisez la CodeDeploy console, vous devez disposer d'un ensemble minimal d'autorisations vous permettant de décrire les autres AWS ressources de votre AWS compte. Pour l'utiliser CodeDeploy dans la CodeDeploy console, vous devez disposer des autorisations des services suivants :

- Amazon EC2 Auto Scaling
- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Gestion des identités et des accès AWS
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon CloudWatch

Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les utilisateurs ayant un rôle dans cette politique IAM. Pour garantir que ces utilisateurs peuvent toujours utiliser la CodeDeploy console, associez également la politique `AWSCodeDeployReadOnlyAccess` gérée au rôle attribué à l'utilisateur, comme décrit dans [AWS politiques gérées \(prédéfinies\) pour CodeDeploy](#).

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' API CodeDeploy API.



## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupsForUser",
 "iam:ListAttachedUserPolicies",
 "iam:ListUserPolicies",
 "iam:GetUser"
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
 },
 {
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam:GetPolicyVersion",
 "iam:GetPolicy",
 "iam:ListAttachedGroupPolicies",
 "iam:ListGroupPolicies",
 "iam:ListPolicyVersions",
 "iam:ListPolicies",
 "iam:ListUsers"
],
 "Resource": "*"
 }
]
}
```

# Résolution des problèmes AWS CodeDeploy d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec CodeDeploy IAM.

## Rubriques

- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes CodeDeploy ressources](#)

### Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à CodeDeploy.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans CodeDeploy. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

### Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes CodeDeploy ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez

spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises CodeDeploy en charge, consultez [Comment AWS CodeDeploy fonctionne avec IAM](#).
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## CodeDeploy référence aux autorisations

Utilisez le tableau suivant lorsque vous configurez des politiques d'accès et rédigez des politiques d'autorisation que vous pouvez associer à une identité IAM (politiques basées sur l'identité). Le tableau répertorie chaque opération d' CodeDeploy API, les actions pour lesquelles vous pouvez accorder des autorisations pour effectuer l'action et le format de l'ARN de la ressource à utiliser pour accorder des autorisations. Vous spécifiez les actions dans le champ `Action` de la politique. Vous spécifiez un ARN, avec ou sans caractère générique (\*) comme valeur de ressource dans le champ `Resource` de la stratégie.

Vous pouvez utiliser des AWS clés de condition larges dans vos CodeDeploy polices pour exprimer des conditions. Pour obtenir la liste complète des touches AWS-wide, consultez la section [Clés disponibles](#) dans le guide de l'utilisateur IAM.

Pour spécifier une action, utilisez le préfixe `codedeploy:` suivi du nom de l'opération d'API (par exemple, `codedeploy:GetApplication` ou `codedeploy:CreateApplication`). Pour spécifier

plusieurs actions dans une même instruction, séparez-les par une virgule (par exemple, "Action": ["codedeploy:action1", "codedeploy:action2"]).

### Utilisation de caractères génériques

Vous pouvez utiliser un caractère générique (\*) dans votre ARN pour spécifier plusieurs actions ou ressources. Par exemple, `codedeploy:*` spécifie toutes les CodeDeploy actions et `codedeploy:Get*` indique toutes les CodeDeploy actions qui commencent par le mot `Get`. Dans l'exemple suivant, il est donné accès à tous les groupes de déploiement dont le nom commence par `West` et qui sont associés à des applications dont le nom débute par `Test`.

```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

Vous pouvez utiliser des caractères génériques avec les ressources suivantes qui sont répertoriées dans le tableau :

- *application-name*
- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

Les caractères génériques ne peuvent pas être utilisés avec *region* ou *account-id*. Pour plus d'informations sur les caractères génériques, consultez la section [Identifiants IAM](#) dans le guide de l'utilisateur IAM.

#### Note

Dans l'ARN de chaque action, un signe deux-points (:) suit la ressource. Vous pouvez également faire suivre la ressource par une barre oblique (/). Pour plus d'informations, voir l'[CodeDeploy exemple ARNs](#).

### CodeDeploy Opérations d'API et autorisations requises pour les actions

#### [AddTagsToOnPremisesInstances](#)

Action : `codedeploy:AddTagsToOnPremisesInstances`

Requise pour ajouter des balises à une ou plusieurs instances sur site.

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [BatchGetApplicationRevisions](#)

Action : `codedeploy:BatchGetApplicationRevisions`

Requête pour obtenir des informations sur plusieurs révisions d'application associées à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### [BatchGetApplications](#)

Action : `codedeploy:BatchGetApplications`

Requête pour obtenir des informations sur plusieurs applications associées à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:*`

### [BatchGetDeploymentGroups](#)

Action : `codedeploy:BatchGetDeploymentGroups`

Requête pour obtenir des informations sur plusieurs groupes de déploiement associés à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [BatchGetDeploymentInstances](#)

Action : `codedeploy:BatchGetDeploymentInstances`

Requête pour obtenir des informations sur une ou plusieurs instances qui font partie d'un groupe de déploiement.

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [BatchGetDeployments](#)

Action : `codedeploy:BatchGetDeployments`

Requête pour obtenir des informations sur plusieurs déploiements associés à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### BatchGetOnPremisesInstances

Action : `codedeploy:BatchGetOnPremisesInstances`

Requête pour obtenir des informations sur une ou plusieurs instances sur site.

Ressource : `arn:aws:codedeploy:region:account-id:*`

### ContinueDeployment

Action : `codedeploy:ContinueDeployment`

Nécessaire lors d'un blue/green déploiement pour commencer à enregistrer des instances dans un environnement de remplacement avec un équilibreur de charge Elastic Load Balancing.

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### CreateApplication

Action : `codedeploy>CreateApplication`

Requête pour créer une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### CreateDeployment<sup>1</sup>

Action : `codedeploy>CreateDeployment`

Requête pour créer un déploiement pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### CreateDeploymentConfig

Action : `codedeploy>CreateDeploymentConfig`

Requête pour créer une configuration de déploiement personnalisée associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

### [CreateDeploymentGroup](#)

Action : `codedeploy:CreateDeploymentGroup`

Requise pour créer un groupe de déploiement pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [DeleteApplication](#)

Action : `codedeploy>DeleteApplication`

Requise pour supprimer une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### [DeleteDeploymentConfig](#)

Action : `codedeploy>DeleteDeploymentConfig`

Requise pour supprimer une configuration de déploiement personnalisée associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

### [DeleteDeploymentGroup](#)

Action : `codedeploy>DeleteDeploymentGroup`

Requise pour supprimer un groupe de déploiement d'une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [DeregisterOnPremisesInstance](#)

Action : `codedeploy:DeregisterOnPremisesInstance`

Requise pour annuler l'enregistrement d'une instance sur site.

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [GetApplication](#)

Action : `codedeploy:GetApplication`

Requête pour obtenir des informations sur une application spécifique associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### [GetApplicationRevision](#)

Action : `codedeploy:GetApplicationRevision`

Requête pour obtenir des informations sur une révision d'une application spécifique associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### [GetDeployment](#)

Action : `codedeploy:GetDeployment`

Requête pour obtenir des informations sur un déploiement spécifique dans un groupe de déploiement pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetDeploymentConfig](#)

Action : `codedeploy:GetDeploymentConfig`

Requête pour obtenir des informations sur une configuration de déploiement spécifique associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

### [GetDeploymentGroup](#)

Action : `codedeploy:GetDeploymentGroup`



Requête pour obtenir des informations sur un groupe de déploiement spécifique pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetDeploymentInstance](#)

Action : `codedeploy:GetDeploymentInstance`

Requête pour obtenir des informations sur une instance spécifique dans un déploiement associé à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetOnPremisesInstance](#)

Action : `codedeploy:GetOnPremisesInstance`

Requête pour obtenir des informations sur une instance sur site spécifique.

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [ListApplicationRevisions](#)

Action : `codedeploy:ListApplicationRevisions`

Requête pour obtenir des informations sur toutes les révisions d'une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:*`

### [ListApplications](#)

Action : `codedeploy:ListApplications`

Requête pour obtenir des informations sur toutes les applications associées à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:*`

### [ListDeploymentConfigs](#)

Action : `codedeploy:ListDeploymentConfigs`

Requête pour obtenir des informations sur toutes les configurations de déploiement associées à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

### [ListDeploymentGroups](#)

Action : `codedeploy:ListDeploymentGroups`

Requête pour obtenir des informations sur tous les groupes de déploiement d'une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

### [ListDeploymentInstances](#)

Action : `codedeploy:ListDeploymentInstances`

Nécessaire pour obtenir des informations sur toutes les instances d'un déploiement associé à l'utilisateur.

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [ListDeployments](#)

Action : `codedeploy:ListDeployments`

Nécessaire pour obtenir des informations sur tous les déploiements vers un groupe de déploiement associé à l'utilisateur, ou pour obtenir tous les déploiements associés à l'utilisateur.

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [ListGitHubAccountTokenNames](#)

Action : `codedeploy:ListGitHubAccountTokenNames`

Nécessaire pour obtenir la liste des noms des connexions enregistrées aux GitHub comptes.

Ressource : `arn:aws:codedeploy:region:account-id:*`

### [ListOnPremisesInstances](#)

Action : `codedeploy:ListOnPremisesInstances`

Requise pour répertorier un ou plusieurs noms d'instance sur site.

Ressource : `arn:aws:codedeploy:region:account-id:*`

### [RegisterApplicationRevision](#)

Action : `codedeploy:RegisterApplicationRevision`

Requise pour enregistrer les informations sur la révision d'une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### [RegisterOnPremisesInstance](#)

Action : `codedeploy:RegisterOnPremisesInstance`

Nécessaire pour enregistrer une instance locale auprès CodeDeploy de.

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [RemoveTagsFromOnPremisesInstances](#)

Action : `codedeploy:RemoveTagsFromOnPremisesInstances`

Requise pour supprimer les balises d'une ou plusieurs instances sur site.

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [SkipWaitTimeForInstanceTermination](#)

Action : `codedeploy:SkipWaitTimeForInstanceTermination`

Nécessaire pour annuler un temps d'attente spécifié et commencer à mettre fin aux instances dans l'environnement d'origine immédiatement après que le trafic a été correctement acheminé dans le cadre d'un déploiement. blue/green

Ressource : `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [StopDeployment](#)

Action : `codedeploy:StopDeployment`

Requise pour arrêter un déploiement en cours dans un groupe de déploiement pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### UpdateApplication<sup>3</sup>

Action : `codedeploy:UpdateApplication`

Requête pour modifier les informations sur une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:application:application-name`

### UpdateDeploymentGroup<sup>3</sup>

Action : `codedeploy:UpdateDeploymentGroup`

Requête pour modifier les informations sur un groupe de déploiement spécifique pour une application associée à l'utilisateur .

Ressource : `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

<sup>1</sup> Lorsque vous spécifiez `CreateDeployment` des autorisations, vous devez également spécifier `GetDeploymentConfig` des autorisations pour la configuration du déploiement `GetApplicationRevision` et/ou `RegisterApplicationRevision` des autorisations pour la révision de l'application.

<sup>2</sup> Valable `ListDeployments` lorsque vous fournissez un groupe de déploiement, mais pas lorsque vous listez tous les déploiements associés à l'utilisateur.

<sup>3</sup> Pour `UpdateApplication`, vous devez disposer `UpdateApplication` d'autorisations pour l'ancien et le nouveau nom de l'application. Pour les actions `UpdateDeploymentGroup` qui impliquent la modification du nom d'un groupe de déploiement, vous devez disposer des autorisations `UpdateDeploymentGroup` pour l'ancien et le nouveau noms de groupe de déploiement.

## Prévention du cas de figure de l'adjoint 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 la confusion des 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 auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés contextuelles de condition SourceAccount globale [aws : SourceArn](#) et [aws :](#) dans les politiques de ressources afin de limiter les autorisations qui CodeDeploy accordent un autre service à la ressource. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws : SourceArn` contient l'ID de compte, la valeur `aws : SourceAccount` et le compte dans la valeur `aws : SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. Utilisez `aws : SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. À utiliser `aws : SourceAccount` si vous souhaitez qu'une ressource de ce compte soit associée à l'utilisation interservices.

Pour les déploiements EC2/on-premises AWS , Lambda et Amazon ECS classiques, la valeur de doit `aws : SourceArn` inclure l'ARN du CodeDeploy groupe de déploiement avec lequel il CodeDeploy est autorisé à assumer le rôle IAM.

Pour les [déploiements bleu/vert Amazon ECS créés par le biais](#) de CloudFormation, la valeur de `aws : SourceArn` doit inclure l' CloudFormation ARN de pile avec lequel CodeDeploy il est autorisé à assumer le rôle IAM.

Le moyen le plus efficace de se protéger contre le problème de confusion des adjoints est d'utiliser la `aws : SourceArn` clé avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet ou si vous spécifiez plusieurs ressources, utilisez des caractères génériques (\*) pour les parties inconnues.

Par exemple, vous pouvez utiliser la politique de confiance suivante dans le cadre d'un déploiement EC2/on-premises, AWS Lambda ou Amazon ECS normal :

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
```

```

 "Principal": {
 "Service": "codedeploy.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "111122223333"
 },
 "ArnLike": {
 "aws:SourceArn": "arn:aws:codedeploy:us-east-1:111122223333:deploymentgroup:myApplication/*"
 }
 }
 }
]
}

```

Pour un [déploiement bleu/vert Amazon ECS créé via CloudFormation](#), vous pouvez utiliser :

JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "codedeploy.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "111122223333"
 },
 "ArnLike": {
 "aws:SourceArn": "arn:aws:cloudformation:us-east-1:111122223333:stack/MyCloudFormationStackName/*"
 }
 }
 }
]
}

```

}

## Connexion et surveillance CodeDeploy

Cette section fournit une vue d'ensemble de la surveillance, de la journalisation et de la réponse aux incidents dans CodeDeploy.

### Audit de toutes les interactions avec CodeDeploy

CodeDeploy est intégré à AWS CloudTrail un service qui capture les appels d'API effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux CodeDeploy dans un compartiment S3 que vous spécifiez. CloudTrail capture les appels d'API depuis la CodeDeploy console AWS CLI, depuis des CodeDeploy commandes via ou CodeDeploy APIs directement. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer à quelle demande a été faite CodeDeploy, l'adresse IP source à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite, etc. Pour en savoir plus CloudTrail, consultez la section [Utilisation des fichiers CloudTrail journaux](#) dans le Guide de AWS CloudTrail l'utilisateur.

Vous pouvez consulter les données du journal créées par un CodeDeploy déploiement en configurant l' CloudWatchagent Amazon pour qu'il affiche les données agrégées dans la CloudWatch console ou en vous connectant à une instance pour consulter le fichier journal. Pour de plus amples informations, veuillez consulter [Envoyer les journaux des CodeDeploy agents à CloudWatch](#).

### Alertes et gestion des incidents

Vous pouvez utiliser Amazon CloudWatch Events pour détecter et réagir aux changements d'état d'une instance ou à un déploiement (un événement) dans le cadre de vos CodeDeploy opérations. Ensuite, en fonction des règles que vous créez, CloudWatch Events invoque une ou plusieurs actions cibles lorsqu'un déploiement ou une instance entre dans l'état que vous spécifiez dans une règle. En fonction du type de modification de l'état, vous pouvez envoyer des notifications, capturer les informations d'état, prendre des mesures correctives, déclencher des événements ou prendre d'autres mesures. Vous pouvez sélectionner les types de cibles suivants lorsque vous utilisez CloudWatch des événements dans le cadre de vos CodeDeploy opérations :

- AWS Lambda fonctions
- Flux Kinesis
- Files d'attente Amazon SQS

- Cibles intégrées (actions CloudWatch d'alarme)
- Rubriques Amazon SNS

Voici quelques cas d'utilisation :

- Utilisez une fonction Lambda pour transmettre une notification à un canal Slack chaque fois que des déploiements échouent.
- Envoyez des données sur les déploiements ou instances à un flux Kinesis pour prendre en charge la surveillance complète et en temps réel de l'état.
- Utilisez des actions CloudWatch d'alarme pour arrêter, mettre fin, redémarrer ou récupérer automatiquement les instances EC2 lorsqu'un événement de déploiement ou d'instance que vous spécifiez se produit.

Pour plus d'informations, consultez la section [Qu'est-ce qu'Amazon CloudWatch Events](#) dans le guide de CloudWatch l'utilisateur Amazon.

## Validation de conformité pour AWS CodeDeploy

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 CodeDeploy

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS 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 à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle 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 CodeDeploy

En tant que service géré, AWS CodeDeploy il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc [Amazon Web Services : présentation des processus de sécurité](#).

Vous utilisez des appels d'API AWS publiés pour accéder CodeDeploy via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Nous recommandons TLS 1.3 ou version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associée à un mandataire IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

# Référence

Référence.

## Rubriques

- [CodeDeploy AppSpec référence de fichier](#)
- [CodeDeploy référence de configuration de l'agent](#)
- [CloudFormation modèles de CodeDeploy référence](#)
- [Utilisation CodeDeploy avec Amazon Virtual Private Cloud](#)
- [CodeDeploy référence du kit de ressources](#)
- [CodeDeploy quotas](#)

## CodeDeploy AppSpec référence de fichier

Cette section est fournie à des fins de référence uniquement. Pour une présentation conceptuelle du AppSpec fichier, voir [Application Specification Files](#).

Le fichier de spécification de l'application (AppSpec fichier) est un fichier au format [YAML](#) ou au format JSON utilisé pour gérer un déploiement. CodeDeploy

### Note

Le AppSpec fichier d'un déploiement EC2/local doit être nommé `appspec.yml`, sauf si vous effectuez un déploiement local. Pour de plus amples informations, veuillez consulter [Création d'un déploiement local](#).

## Rubriques

- [AppSpec fichiers sur une plateforme de calcul Amazon ECS](#)
- [AppSpec fichiers sur une plate-forme AWS Lambda informatique](#)
- [AppSpec fichiers sur une plate-forme informatique EC2/sur site](#)
- [AppSpec Structure du fichier](#)
- [AppSpec Exemple de fichier](#)
- [AppSpec Espacement des fichiers](#)

- [Validez votre AppSpec fichier et son emplacement](#)

## AppSpec fichiers sur une plateforme de calcul Amazon ECS

Pour les applications de plate-forme de calcul Amazon ECS, le AppSpec fichier est utilisé CodeDeploy pour déterminer :

- Votre fichier de définition de tâche Amazon ECS. Ceci est spécifié avec son ARN dans l'`TaskDefinitionInstruction` du AppSpec fichier.
- Le conteneur et le port de votre ensemble de tâches de remplacement où votre Application Load Balancer ou Network Load Balancer redirige le trafic lors d'un déploiement. Ceci est spécifié dans les `LoadBalancerInfo` instructions du AppSpec fichier.
- Informations facultatives concernant votre service Amazon ECS, telles que la version de plate-forme sur laquelle il s'exécute, ses sous-réseaux et ses groupes de sécurité.
- Fonctions Lambda facultatives à exécuter pendant les hooks correspondant aux événements du cycle de vie lors d'un déploiement Amazon ECS. Pour de plus amples informations, veuillez consulter [AppSpec section « hooks » pour un déploiement Amazon ECS](#).

## AppSpec fichiers sur une plate-forme AWS Lambda informatique

Pour les applications de plate-forme de calcul AWS Lambda, le AppSpec fichier est utilisé pour CodeDeploy déterminer :

- Quelle version de la fonction Lambda déployer.
- Quelles fonctions Lambda utiliser comme tests de validation.

Un AppSpec fichier peut être au format YAML ou au format JSON. Vous pouvez également saisir le contenu d'un AppSpec fichier directement dans CodeDeploy la console lorsque vous créez un déploiement.

## AppSpec fichiers sur une plate-forme informatique EC2/sur site

Si votre application utilise la plate-forme de calcul EC2/on-premises, le AppSpec fichier doit être un fichier au format YAML nommé `appspec.yml` et il doit être placé à la racine de la structure de répertoires du code source de l'application. Dans le cas contraire, les déploiements échouent. Il est utilisé CodeDeploy pour déterminer :

- Ce qu'il doit installer sur vos instances à partir de la révision de votre application dans Amazon S3 ou GitHub.
- Quels hooks d'événement de cycle de vie exécuter en réponse à des événements de cycle de vie du déploiement.

Une fois AppSpec le fichier terminé, vous le regroupez, avec le contenu à déployer, dans un fichier d'archive (zip, tar ou tar compressé). Pour de plus amples informations, veuillez consulter [Travailler avec les révisions d'applications pour CodeDeploy](#).

 Note

Les formats de fichier d'archive tar et tar compressé (.tar et .tar.gz) ne sont pas pris en charge pour les instances Windows Server.

Une fois que vous avez un fichier d'archive groupé (connu CodeDeploy sous le nom de révision), vous le chargez dans un bucket Amazon S3 ou un référentiel Git. Ensuite, vous utilisez CodeDeploy pour déployer la révision. Pour obtenir des instructions, veuillez consulter [Créez un déploiement avec CodeDeploy](#).

Le fichier appspec.yml pour un déploiement de plate-forme de calcul EC2/sur site est enregistré dans le répertoire racine de votre révision. Pour plus d'informations, consultez [Ajouter un AppSpec fichier pour un déploiement EC2/sur site](#) et [Planifier une révision pour CodeDeploy](#).

## AppSpec Structure du fichier

Voici la structure de haut niveau d'un AppSpec fichier utilisé pour les déploiements sur les plateformes de calcul AWS Lambda et EC2/on-premises.

Une valeur d'un AppSpec fichier au format YAML qui est une chaîne ne doit pas être placée entre guillemets (« ») sauf indication contraire.

## AppSpec structure de fichiers pour les déploiements Amazon ECS

### Note

Ce AppSpec fichier est écrit en YAML, mais vous pouvez utiliser la même structure pour en écrire un en JSON. Dans un AppSpec fichier au format JSON, une chaîne est toujours placée entre guillemets (« »).

```
version: 0.0
resources:
 ecs-service-specifications
hooks:
 deployment-lifecycle-event-mappings
```

Dans cette structure :

### Version

Cette section indique la version du AppSpec fichier. Ne modifiez pas cette valeur. C'est obligatoire. Actuellement, la seule valeur autorisée est **0.0**. Il est réservé CodeDeploy pour une utilisation future.

Spécifiez version avec une chaîne.

### resources

Cette section fournit des informations sur l'application Amazon ECS à déployer.

Pour de plus amples informations, veuillez consulter [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

### hooks

Cette section décrit les fonctions Lambda à exécuter lors d'événements spécifiques du cycle de vie du déploiement afin de valider le déploiement.

Pour de plus amples informations, veuillez consulter [Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS](#).

## AppSpec structure de fichiers pour les AWS déploiements Lambda

### Note

Ce AppSpec fichier est écrit en YAML, mais vous pouvez utiliser la même structure pour écrire un AppSpec fichier pour un déploiement Lambda en JSON. Dans un AppSpec fichier au format JSON, une chaîne est toujours placée entre guillemets (« »).

```
version: 0.0
resources:
 lambda-function-specifications
hooks:
 deployment-lifecycle-event-mappings
```

Dans cette structure :

### Version

Cette section indique la version du AppSpec fichier. Ne modifiez pas cette valeur. C'est obligatoire. Actuellement, la seule valeur autorisée est **0.0**. Il est réservé CodeDeploy pour une utilisation future.

Spécifiez version avec une chaîne.

### resources

Cette section fournit des informations sur la fonction Lambda à déployer.

Pour de plus amples informations, veuillez consulter [AppSpec section « ressources » \(Amazon ECS et AWS Lambda déploiements uniquement\)](#).

### hooks

Cette section décrit les fonctions Lambda à exécuter lors d'événements spécifiques du cycle de vie du déploiement afin de valider le déploiement.

Pour de plus amples informations, veuillez consulter [AppSpec section « crochets »](#).

## AppSpec structure de fichiers pour les déploiements EC2/sur site

```
version: 0.0
```

```
os: operating-system-name
files:
 source-destination-files-mappings
permissions:
 permissions-specifications
hooks:
 deployment-lifecycle-event-mappings
```

Dans cette structure :

## Version

Cette section indique la version du AppSpec fichier. Ne modifiez pas cette valeur. C'est obligatoire. Actuellement, la seule valeur autorisée est **0.0**. Il est réservé CodeDeploy pour une utilisation future.

Spécifiez version avec une chaîne.

## os

Cette section spécifie la valeur du système d'exploitation de l'instance sur laquelle vous déployez. C'est obligatoire. Les valeurs suivantes peuvent être spécifiées :

- linux — L'instance est une instance Amazon Linux, Ubuntu Server ou RHEL.
- windows — L'instance est une instance Windows Server.

Spécifiez os avec une chaîne.

## files

Cette section spécifie les noms des fichiers qui doivent être copiés sur l'instance au cours de l'événement Install du déploiement.

Pour de plus amples informations, veuillez consulter [AppSpec section « fichiers » \(déploiements EC2/sur site uniquement\)](#).

## permissions

Cette section spécifie les autorisations spéciales éventuelles qui doivent être appliquées aux fichiers de la section `files` lorsqu'ils sont copiés vers l'instance. Cette section s'applique uniquement aux instances Amazon Linux, Ubuntu Server et Red Hat Enterprise Linux (RHEL).

Pour plus d'informations, voir, [AppSpec section « autorisations » \(déploiements EC2/sur site uniquement\)](#).

## hooks

Cette section spécifie les scripts à exécuter lors d'événements spécifiques de cycle de vie de déploiement, au cours du déploiement.

Pour de plus amples informations, veuillez consulter [AppSpec section « crochets »](#).

## Rubriques

- [AppSpec section « fichiers » \(déploiements EC2/sur site uniquement\)](#)
- [AppSpec section « ressources » \(Amazon ECS et AWS Lambda déploiements uniquement\)](#)
- [AppSpec section « autorisations » \(déploiements EC2/sur site uniquement\)](#)
- [AppSpec section « crochets »](#)

## AppSpec section « fichiers » (déploiements EC2/sur site uniquement)

Fournit des informations CodeDeploy sur les fichiers issus de la révision de votre application qui doivent être installés sur l'instance lors de l'événement Install du déploiement. Cette section est obligatoire uniquement si vous copiez des fichiers à partir de votre révision dans des emplacements sur l'instance durant le déploiement.

Cette section a la structure suivante :

```
files:
 - source: source-file-location-1
 destination: destination-file-location-1
 file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

Plusieurs paires source et destination peuvent être définies.

L'instruction source identifie un fichier ou un répertoire de votre révision à copier sur l'instance :

- Si source fait référence à un fichier, seuls les fichiers spécifiés sont copiés sur l'instance.
- Si source fait référence à un répertoire, tous les fichiers figurant dans le répertoire sont copiés sur l'instance.
- S'il s'agit d'une barre oblique unique («/» pour les instances Amazon Linux, RHEL et Ubuntu Server, ou «\» pour les instances Windows Server), tous les fichiers de votre révision sont copiés sur l'instance.



Les chemins utilisés dans `source` sont relatifs au `appspec.yml` fichier, qui doit se trouver à la racine de votre révision. Pour plus de détails sur la structure de fichier d'une révision, voir [Planifier une révision pour CodeDeploy](#).

L'instruction `destination` identifie l'emplacement sur l'instance où les fichiers doivent être copiés. Il doit s'agir d'un chemin entièrement qualifié tel que `/root/destination/directory` (sous Linux, RHEL et Ubuntu) ou `c:\destination\folder` (sous Windows).

Les instructions `source` et `destination` sont spécifiées chacune avec une chaîne.

L'`file_exists_behavior` instruction est facultative et indique comment CodeDeploy gère les fichiers qui existent déjà dans un emplacement cible de déploiement mais qui ne faisaient pas partie du déploiement réussi précédent. Ce paramètre peut prendre l'une des valeurs suivantes :

- **INTERDIRE** : le déploiement échoue. C'est également le comportement par défaut si aucune option n'est spécifiée.
- **REEMPLACER** : La version du fichier issue de la révision de l'application en cours de déploiement remplace la version déjà présente sur l'instance.
- **CONSERVER** : La version du fichier déjà présente sur l'instance est conservée et utilisée dans le cadre du nouveau déploiement.

Lorsque vous utilisez le `file_exists_behavior` paramètre, sachez que celui-ci :

- ne peut être spécifié qu'une seule fois et s'applique à tous les fichiers et répertoires répertoriés ci-dessous `files:`.
- a priorité sur l'`--file-exists-behavior` AWS CLI option et l'option `fileExistsBehavior` API (qui sont également facultatives).

Voici un exemple de `files` section pour une instance Amazon Linux, Ubuntu Server ou RHEL.

```
files:
 - source: Config/config.txt
 destination: /webapps/Config
 - source: source
 destination: /webapps/myApp
```

Dans cet exemple, les deux opérations suivantes sont effectuées au cours de l'événement `Install` :

1. Copie du fichier Config/config.txt de votre révision vers le chemin /webapps/Config/config.txt sur l'instance.
2. Copie récursive de tous les fichiers du répertoire source de votre révision vers le répertoire /webapps/myApp sur l'instance.

## Exemples de sections « Fichiers »

Les exemples suivants montrent comment spécifier la section `files`. Bien que ces exemples décrivent les structures de fichiers et de répertoires (dossiers) de Windows Server, ils peuvent facilement être adaptés aux instances Amazon Linux, Ubuntu Server et RHEL.

### Note

Seuls les déploiements EC2/sur site utilisent cette section. `files` Elle ne s'applique pas aux AWS déploiements Lambda.

Pour les exemples suivants, nous supposons que ces fichiers apparaissent dans le groupe à la racine de source :

- appspec.yml
- my-file.txt
- my-file-2.txt
- my-file-3.txt

```
1) Copy only my-file.txt to the destination folder c:\temp.
#
files:
 - source: .\my-file.txt
 destination: c:\temp
#
Result:
c:\temp\my-file.txt
#

#
2) Copy only my-file-2.txt and my-file-3.txt to the destination folder c:\temp.
#
```

```

files:
 - source: my-file-2.txt
 destination: c:\temp
 - source: my-file-3.txt
 destination: c:\temp
#
Result:
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt
#

#
3) Copy my-file.txt, my-file-2.txt, and my-file-3.txt (along with the appspec.yml
file) to the destination folder c:\temp.
#
files:
 - source: \
 destination: c:\temp
#
Result:
c:\temp\appspec.yml
c:\temp\my-file.txt
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt

```

Pour les exemples suivants, nous supposons que `appspec.yml` apparaît dans le groupe à la racine de source avec un dossier nommé `my-folder` contenant trois fichiers :

- `appspec.yml`
- `my-folder\my-file.txt`
- `my-folder\my-file-2.txt`
- `my-folder\my-file-3.txt`

```

4) Copy the 3 files in my-folder (but do not copy my-folder itself) to the
destination folder c:\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp
#
Result:

```

```
c:\temp\my-file.txt
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt
#

#
5) Copy my-folder and its 3 files to my-folder within the destination folder c:\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp\my-folder
#
Result:
c:\temp\my-folder\my-file.txt
c:\temp\my-folder\my-file-2.txt
c:\temp\my-folder\my-file-3.txt
#

#
6) Copy the 3 files in my-folder to other-folder within the destination folder c:
\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp\other-folder
#
Result:
c:\temp\other-folder\my-file.txt
c:\temp\other-folder\my-file-2.txt
c:\temp\other-folder\my-file-3.txt
#

#
7) Copy only my-file-2.txt and my-file-3.txt to my-folder within the destination
folder c:\temp.
#
files:
 - source: .\my-folder\my-file-2.txt
 destination: c:\temp\my-folder
 - source: .\my-folder\my-file-3.txt
 destination: c:\temp\my-folder
#
Result:
c:\temp\my-folder\my-file-2.txt
```

```
c:\temp\my-folder\my-file-3.txt
#

#
8) Copy only my-file-2.txt and my-file-3.txt to other-folder within the destination
 folder c:\temp.
#
files:
 - source: .\my-folder\my-file-2.txt
 destination: c:\temp\other-folder
 - source: .\my-folder\my-file-3.txt
 destination: c:\temp\other-folder
#
Result:
c:\temp\other-folder\my-file-2.txt
c:\temp\other-folder\my-file-3.txt
#

#
9) Copy my-folder and its 3 files (along with the appspec.yml file) to the
 destination folder c:\temp. If any of the files already exist on the instance,
 overwrite them.
#
files:
 - source: \
 destination: c:\temp
file_exists_behavior: OVERWRITE
#
Result:
c:\temp\appspec.yml
c:\temp\my-folder\my-file.txt
c:\temp\my-folder\my-file-2.txt
c:\temp\my-folder\my-file-3.txt
```

## AppSpec section « ressources » (Amazon ECS et AWS Lambda déploiements uniquement)

Le contenu de la 'resources' section du AppSpec fichier varie en fonction de la plate-forme de calcul de votre déploiement. La 'resources' section relative au déploiement d'Amazon ECS contient la définition de votre tâche Amazon ECS, le conteneur et le port pour acheminer le trafic vers votre ensemble de tâches Amazon ECS mis à jour, ainsi que d'autres informations facultatives. La

'resources' section relative à un AWS Lambda déploiement contient le nom, l'alias, la version actuelle et la version cible d'une fonction Lambda.

## Rubriques

- [AppSpec section « ressources » pour les déploiements AWS Lambda](#)
- [AppSpec section « ressources » pour les déploiements Amazon ECS](#)

### AppSpec section « ressources » pour les déploiements AWS Lambda

La 'resources' section indique la fonction Lambda à déployer et présente la structure suivante :

YAML :

```
resources:
 - name-of-function-to-deploy:
 type: "AWS::Lambda::Function"
 properties:
 name: name-of-lambda-function-to-deploy
 alias: alias-of-lambda-function-to-deploy
 currentversion: version-of-the-lambda-function-traffic-currently-points-to
 targetversion: version-of-the-lambda-function-to-shift-traffic-to
```

JSON:

```
"resources": [
 {
 "name-of-function-to-deploy" {
 "type": "AWS::Lambda::Function",
 "properties": {
 "name": "name-of-lambda-function-to-deploy",
 "alias": "alias-of-lambda-function-to-deploy",
 "currentversion": "version-of-the-lambda-function-traffic-currently-points-to",
 "targetversion": "version-of-the-lambda-function-to-shift-traffic-to"
 }
 }
 }
]
```

Chaque propriété est spécifiée avec une chaîne.

- **name** : obligatoire. Il s'agit du nom de la fonction Lambda à déployer.
- **alias** : obligatoire. Il s'agit du nom de l'alias de la fonction Lambda.
- **currentversion** : obligatoire. Il s'agit de la version de la fonction Lambda vers laquelle le trafic pointe actuellement. Cette valeur doit être un entier positif valide.
- **targetversion** : obligatoire. Il s'agit de la version de la fonction Lambda vers laquelle le trafic est transféré. Cette valeur doit être un entier positif valide.

## AppSpec section « ressources » pour les déploiements Amazon ECS

La 'resources' section indique le service Amazon ECS à déployer et présente la structure suivante :

YAML :

```
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
 TaskDefinition: "task-definition-arn"
 LoadBalancerInfo:
 ContainerName: "ecs-container-name"
 ContainerPort: "ecs-application-port"
Optional properties
 PlatformVersion: "ecs-service-platform-version"
 NetworkConfiguration:
 AwsVpcConfiguration:
 Subnets: ["ecs-subnet-1", "ecs-subnet-n"]
 SecurityGroups: ["ecs-security-group-1", "ecs-security-group-n"]
 AssignPublicIp: "ENABLED | DISABLED"
 CapacityProviderStrategy:
 - Base: integer
 CapacityProvider: "capacityProviderA"
 Weight: integer
 - Base: integer
 CapacityProvider: "capacityProviderB"
 Weight: integer
```

JSON:

```
"Resources": [
 {
```

```

"TargetService": {
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "task-definition-arn",
 "LoadBalancerInfo": {
 "ContainerName": "ecs-container-name",
 "ContainerPort": "ecs-application-port"
 },
 "PlatformVersion": "ecs-service-platform-version",
 "NetworkConfiguration": {
 "AwsVpcConfiguration": {
 "Subnets": [
 "ecs-subnet-1",
 "ecs-subnet-n"
],
 "SecurityGroups": [
 "ecs-security-group-1",
 "ecs-security-group-n"
],
 "AssignPublicIp": "ENABLED | DISABLED"
 }
 },
 "CapacityProviderStrategy": [
 {
 "Base": integer,
 "CapacityProvider": "capacityProviderA",
 "Weight": integer
 },
 {
 "Base": integer,
 "CapacityProvider": "capacityProviderB",
 "Weight": integer
 }
]
 }
}
]

```

Chaque propriété est spécifiée par une chaîne à l'exception de `ContainerPort`, qui est un nombre.

- **TaskDefinition** : obligatoire. Il s'agit de la définition de tâche à déployer pour le service Amazon ECS. Elle est spécifiée par l'ARN de la définition de tâche. Le format ARN est le suivant :



arn:aws:ecs:*aws-region*:*account-id*:task-definition/*task-definition-family*:*task-definition-revision*. Pour plus d'informations, consultez [Amazon Resource Names \(ARNs\) et espaces AWS de noms de services](#).

 Note

La *:task-definition-revision* partie de l'ARN est facultative. S'il est omis, Amazon ECS utilise la dernière révision ACTIVE de la définition de tâche.

- **ContainerName** : obligatoire. Il s'agit du nom du conteneur Amazon ECS qui contient votre application Amazon ECS. Il doit s'agir d'un conteneur spécifié dans votre définition de tâche Amazon ECS.
- **ContainerPort** : obligatoire. Il s'agit du port du conteneur vers lequel le trafic sera acheminé.
- **PlatformVersion**: Facultatif. Version de plateforme des tâches Fargate dans le service Amazon ECS déployé. Pour plus d'informations, consultez [Versions de plateforme AWS Fargate](#). S'il n'est pas spécifié, LATEST il est utilisé par défaut.
- **NetworkConfiguration**: Facultatif. Sous **AwsVpcConfiguration**, vous pouvez spécifier les valeurs suivantes. Pour plus d'informations, consultez [AwsVpcConfiguration](#) le manuel Amazon ECS Container Service API Reference.
  - **Subnets**: Facultatif. Liste séparée par des virgules d'un ou de plusieurs sous-réseaux de votre service Amazon ECS.
  - **SecurityGroups**: Facultatif. Liste séparée par des virgules d'un ou de plusieurs groupes de sécurité dans votre Amazon Elastic Container Service.
  - **AssignPublicIp**: Facultatif. Chaîne qui indique si l'interface Elastic network de votre service Amazon ECS reçoit une adresse IP publique. Les valeurs valides sont ENABLED et DISABLED.

 Note

Sous **NetworkConfiguration**, tous les paramètres doivent être spécifiés ou aucun. Par exemple, si vous souhaitez spécifier **Subnets**, vous devez également spécifier **SecurityGroups** et **AssignPublicIp**. Si aucun n'est spécifié, CodeDeploy utilise les paramètres actuels du réseau Amazon ECS.

- **CapacityProviderStrategy**: Facultatif. Liste des fournisseurs de capacité Amazon ECS que vous souhaitez utiliser pour votre déploiement. Pour plus d'informations, consultez les [fournisseurs de capacité Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Pour chaque fournisseur de capacité, vous pouvez définir les paramètres suivants. Pour plus de détails sur ces paramètres, voir [AWS::ECS::ServiceCapacityProviderStrategyItem](#) le guide de AWS CloudFormation l'utilisateur

- **Base:** Facultatif. La valeur de base indique le nombre minimum de tâches à exécuter sur le fournisseur de capacité spécifié. Une base ne peut être définie que pour un seul fournisseur de capacité dans une stratégie de fournisseur de capacité. Si aucune valeur n'est spécifiée, la valeur par défaut de 0 est utilisée.
- **CapacityProvider:** Facultatif. Nom abrégé du fournisseur de capacité. Exemple : `CapacityProvidera`
- **Weight:** Facultatif.

La valeur de poids indique le pourcentage relatif du nombre total de tâches lancées devant utiliser le fournisseur de capacité spécifié. La valeur `weight` est prise en considération une fois que la valeur `base`, si elle est définie, est satisfaite.

Si aucune valeur `0` n'est spécifiée, la valeur par défaut de `weight` est utilisée. Lorsque plusieurs fournisseurs de capacité sont spécifiés dans le cadre d'une stratégie de fournisseur de capacité, au moins l'un des fournisseurs de capacité doit disposer d'une valeur de pondération supérieure à zéro ; les fournisseurs de capacité dont le poids est égal à `0` ne seront pas utilisés afin de placer des tâches. Si vous spécifiez, dans une stratégie, plusieurs fournisseurs de capacité qui possèdent tous un poids de `0`, toutes les actions `RunTask` ou `CreateService` utilisant la stratégie de fournisseur de capacité échoueront.

Un exemple de scénario d'utilisation des pondérations consiste à définir une stratégie contenant deux fournisseurs de capacité qui possèdent un poids de 1. Puis, lorsque la base est satisfaite, les tâches sont réparties équitablement entre les deux fournisseurs de capacité. Dans la même logique, si vous spécifiez un poids de 1 pour `capacityProviderA` et un poids de 4 pour `capacityProviderB`, pour chaque tâche exécutée avec `capacityProviderA`, quatre tâches utiliseront `capacityProviderB`.

## AppSpec section « autorisations » (déploiements EC2/sur site uniquement)

La 'permissions' section indique comment les autorisations spéciales, le cas échéant, doivent être appliquées aux fichiers et directories/folders à la 'files' section après leur copie sur l'instance. Vous pouvez spécifier plusieurs instructions `object`. Cette section est facultative. Elle s'applique uniquement aux instances Amazon Linux, Ubuntu Server et RHEL.

 Note

La 'permissions' section est uniquement utilisée pour les déploiements EC2/sur site. Il n'est pas utilisé pour les AWS déploiements Lambda ou Amazon ECS.

Cette section a la structure suivante :

```
permissions:
- object: object-specification
 pattern: pattern-specification
 except: exception-specification
 owner: owner-account-name
 group: group-name
 mode: mode-specification
 acls:
 - acls-specification
 context:
 user: user-specification
 type: type-specification
 range: range-specification
 type:
 - object-type
```

Les instructions sont les suivantes :

- **object** : obligatoire. Il s'agit d'un ensemble d'objets de système de fichiers (fichiers ou répertoires/dossiers) auxquels les autorisations spécifiées sont appliquées après la copie des objets de système de fichiers sur l'instance.

Spécifiez **object** avec une chaîne.

- **pattern** : facultatif. Spécifie un modèle pour appliquer des autorisations. Si elles ne sont pas spécifiées ou si elles sont spécifiées avec les caractères spéciaux "\*\*\*", les autorisations sont appliquées à tous les fichiers ou répertoires correspondants, selon leur type.

Spécifiez **pattern** avec une chaîne placée entre guillemets ("").

- **except** : facultatif. Spécifie les fichiers ou répertoires qui sont des exceptions par rapport à **pattern**

Spécifiez **except** avec une liste de chaînes séparées par des virgules entre crochets.

- **owner** : facultatif. Nom du propriétaire de `object`. Si cette instruction n'est pas spécifiée, tous les propriétaires existants appliqués à la structure de fichiers ou de répertoires/dossiers d'origine restent inchangés après l'opération de copie.

Spécifiez `owner` avec une chaîne.

- **group** : facultatif. Nom du groupe pour `object`. Si cette instruction n'est pas spécifiée, tous les groupes existants appliqués à la structure de fichiers ou de répertoires/dossiers d'origine restent inchangés après l'opération de copie.

Spécifiez `group` avec une chaîne.

- **mode** : facultatif. Une valeur numérique spécifiant les autorisations à `object` appliquer. Le réglage du mode suit la syntaxe de la commande Linux `chmod`.

#### Important

Si la valeur inclut un zéro initial, vous devez l'entourer de guillemets doubles ou supprimer le zéro initial afin qu'il ne reste que trois chiffres.

#### Note

La notation symbolique telle que celle-ci n'`u+x`est pas prise en charge pour le mode paramètre.

Exemples :

- `mode : "0644"` donne des autorisations de lecture et d'écriture au propriétaire de l'objet (6), des autorisations de lecture seule au groupe (4) et des autorisations de lecture seule à tous les autres utilisateurs (4).
- `mode : 644` accorde les mêmes autorisations que `mode : "0644"`.
- `mode : 4755` définit l'attribut `setuid` (4), donne des autorisations de contrôle complètes au propriétaire (7), donne des autorisations de lecture et d'exécution au groupe (5) et donne des autorisations de lecture et d'exécution à tous les autres utilisateurs (5).

Pour plus d'exemples, consultez la documentation de la commande Linux `chmod`.

Si aucun mode n'est spécifié, tous les modes existants appliqués à la structure de fichier ou de dossier d'origine restent inchangés après l'opération de copie.

- `acls` : facultatif. Liste de chaînes de caractères représentant une ou plusieurs entrées de liste de contrôle d'accès (ACL) appliquées à `object`. Par exemple, `u:bob:rw` représente des autorisations de lecture et d'écriture pour l'utilisateur **bob**. (Pour plus d'exemples, consultez les exemples de formats d'entrées ACL dans la documentation relative à la commande Linux `setfacl`.) Vous pouvez spécifier plusieurs entrées ACL. Si `acls` ce n'est pas spécifié, tout élément existant ACLs appliqué au fichier ou à la directory/folder structure d'origine reste inchangé après l'opération de copie. Elles remplacent toutes celles qui existent déjà ACLs.

Spécifiez `acls` avec un tiret (-) suivi d'un espace, puis d'une chaîne (par exemple, - `u:jane:rw`). Si vous avez plusieurs listes ACL, chacune est spécifiée sur une ligne distincte.

 Note

La définition d'utilisateurs anonymes, de groupes anonymes ou d'autres entrées ACL similaires entraîne l'échec du AppSpec fichier. Utilisez `mode` pour spécifier ces types d'autorisations à la place.

- `context` : facultatif. Pour les instances compatibles Security-Enhanced Linux (SELinux), liste d'étiquettes contextuelles relatives à la sécurité à appliquer aux objets copiés. Les étiquettes sont spécifiées en tant que clés contenant `user`, `type` et `range`. (Pour plus d'informations, consultez la SELinux documentation.) Chaque clé est entrée avec une chaîne. Si elles ne sont pas spécifiées, toutes les étiquettes existantes appliquées au fichier ou à la directory/folder structure d'origine restent inchangées après l'opération de copie.
  - `user` : facultatif. L' SELinux utilisateur.
  - `type` : facultatif. Le nom du SELinux type.
  - `range` : facultatif. Le spécificateur SELinux de plage. Cette instruction n'a aucun effet, sauf si Multi-Level Security (MLS) et Multi-Category Security (MCS) sont activés sur la machine. Si ces options ne sont pas activées, `range` prend par défaut la valeur **s0**.

Spécifiez `context` avec une chaîne (par exemple, `user: unconfined_u`). Chaque instruction `context` est spécifiée sur une ligne distincte.

- `type` : facultatif. Types d'objets auxquels appliquer les autorisations spécifiées. `type` est une chaîne qui peut être définie sur **file** ou **directory**. Si **file** est spécifié, les autorisations sont appliquées uniquement aux fichiers contenus immédiatement dans `object` après l'opération de

copie (et non à object proprement dit). Si cette **directory** option est spécifiée, les autorisations sont appliquées de manière réursive à tous directories/folders ceux qui se trouvent n'importe où object après l'opération de copie (mais pas à object elle-même).

Spécifiez type avec un tiret (-) suivi d'un espace, puis d'une chaîne (par exemple, - file).

### Exemple de section « Autorisations »

L'exemple suivant montre comment spécifier la section 'permissions' avec les instructions object, pattern, except, owner, mode et type. Cet exemple s'applique uniquement aux instances Amazon Linux, Ubuntu Server et RHEL. Dans cet exemple, supposez que les fichiers et dossiers suivants sont copiés vers l'instance dans la hiérarchie :

```
/tmp
|-- my-app
 |-- my-file-1.txt
 |-- my-file-2.txt
 |-- my-file-3.txt
 |-- my-folder-1
 |-- my-file-4.txt
 |-- my-file-5.txt
 |-- `-- my-file-6.txt
 |-- my-folder-2
 |-- my-file-7.txt
 |-- my-file-8.txt
 |-- my-file-9.txt
 |-- my-folder-3
```

Le AppSpec fichier suivant indique comment définir des autorisations sur ces fichiers et dossiers une fois qu'ils ont été copiés :

```
version: 0.0
os: linux
Copy over all of the folders and files with the permissions they
were originally assigned.
files:
 - source: ./my-file-1.txt
 destination: /tmp/my-app
 - source: ./my-file-2.txt
 destination: /tmp/my-app
 - source: ./my-file-3.txt
```

```
 destination: /tmp/my-app
- source: ./my-folder-1
 destination: /tmp/my-app/my-folder-1
- source: ./my-folder-2
 destination: /tmp/my-app/my-folder-2
1) For all of the files in the /tmp/my-app folder ending in -3.txt
(for example, just my-file-3.txt), owner = adm, group = wheel, and
mode = 464 (-r--rw-r--).
permissions:
- object: /tmp/my-app
 pattern: "*-3.txt"
 owner: adm
 group: wheel
 mode: 464
 type:
 - file
2) For all of the files ending in .txt in the /tmp/my-app
folder, but not for the file my-file-3.txt (for example,
just my-file-1.txt and my-file-2.txt),
owner = ec2-user and mode = 444 (-r--r--r--).
- object: /tmp/my-app
 pattern: "*.txt"
 except: [my-file-3.txt]
 owner: ec2-user
 mode: 444
 type:
 - file
3) For all the files in the /tmp/my-app/my-folder-1 folder except
for my-file-4.txt and my-file-5.txt, (for example,
just my-file-6.txt), owner = operator and mode = 646 (-rw-r--rw-).
- object: /tmp/my-app/my-folder-1
 pattern: "***"
 except: [my-file-4.txt, my-file-5.txt]
 owner: operator
 mode: 646
 type:
 - file
4) For all of the files that are immediately under
the /tmp/my-app/my-folder-2 folder except for my-file-8.txt,
(for example, just my-file-7.txt and
my-file-9.txt), owner = ec2-user and mode = 777 (-rwxrwxrwx).
- object: /tmp/my-app/my-folder-2
 pattern: "***"
 except: [my-file-8.txt]
```

```

 owner: ec2-user
 mode: 777
 type:
 - file
5) For all folders at any level under /tmp/my-app that contain
the name my-folder but not
/tmp/my-app/my-folder-2/my-folder-3 (for example, just
/tmp/my-app/my-folder-1 and /tmp/my-app/my-folder-2),
owner = ec2-user and mode = 555 (dr-xr-xr-x).
- object: /tmp/my-app
 pattern: "*my-folder*"
 except: [tmp/my-app/my-folder-2/my-folder-3]
 owner: ec2-user
 mode: 555
 type:
 - directory
6) For the folder /tmp/my-app/my-folder-2/my-folder-3,
group = wheel and mode = 564 (dr-xrw-r--).
- object: /tmp/my-app/my-folder-2/my-folder-3
 group: wheel
 mode: 564
 type:
 - directory

```

Les autorisations obtenues sont les suivantes :

```

-r--r--r-- ec2-user root my-file-1.txt
-r--r--r-- ec2-user root my-file-2.txt
-r--rw-r-- adm wheel my-file-3.txt

dr-xr-xr-x ec2-user root my-folder-1
-rw-r--r-- root root my-file-4.txt
-rw-r--r-- root root my-file-5.txt
-rw-r--rw- operator root my-file-6.txt

dr-xr-xr-x ec2-user root my-folder-2
-rwxrwxrwx ec2-user root my-file-7.txt
-rw-r--r-- root root my-file-8.txt
-rwxrwxrwx ec2-user root my-file-9.txt

dr-xrw-r-- root wheel my-folder-3

```



L'exemple suivant montre comment spécifier la section 'permissions' avec l'ajout des instructions acls et context. Cet exemple s'applique uniquement aux instances Amazon Linux, Ubuntu Server et RHEL.

```
permissions:
 - object: /var/www/html/WordPress
 pattern: "***"
 except: [/var/www/html/WordPress/ReadMe.txt]
 owner: bob
 group: writers
 mode: 644
 acls:
 - u:mary:rw
 - u:sam:rw
 - m::rw
 context:
 user: unconfined_u
 type: httpd_sys_content_t
 range: s0
 type:
 - file
```

## AppSpec section « crochets »

Le contenu de la 'hooks' section du AppSpec fichier varie en fonction de la plate-forme de calcul utilisée pour votre déploiement. La 'hooks' section relative à un déploiement EC2/sur site contient des mappages qui relient les hooks d'événements du cycle de vie du déploiement à un ou plusieurs scripts. La 'hooks' section relative à un déploiement Lambda ou Amazon ECS indique les fonctions de validation Lambda à exécuter lors d'un événement du cycle de vie du déploiement. Si aucun hook d'événement n'est présent, aucune opération n'est exécutée pour cet événement. Cette section est obligatoire uniquement si vous exécutez des scripts ou des fonctions de validation Lambda dans le cadre du déploiement.

### Rubriques

- [AppSpec section « hooks » pour un déploiement Amazon ECS](#)
- [AppSpec section « hooks » pour un déploiement AWS Lambda](#)
- [AppSpec section « hooks » pour un déploiement EC2/sur site](#)

## AppSpec section « hooks » pour un déploiement Amazon ECS

### Rubriques

- [Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS](#)
- [Exécutez l'ordre des hooks dans un déploiement Amazon ECS.](#)
- [Structure de la section « crochets »](#)
- [Exemple de fonction « hooks » Lambda](#)

### Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS

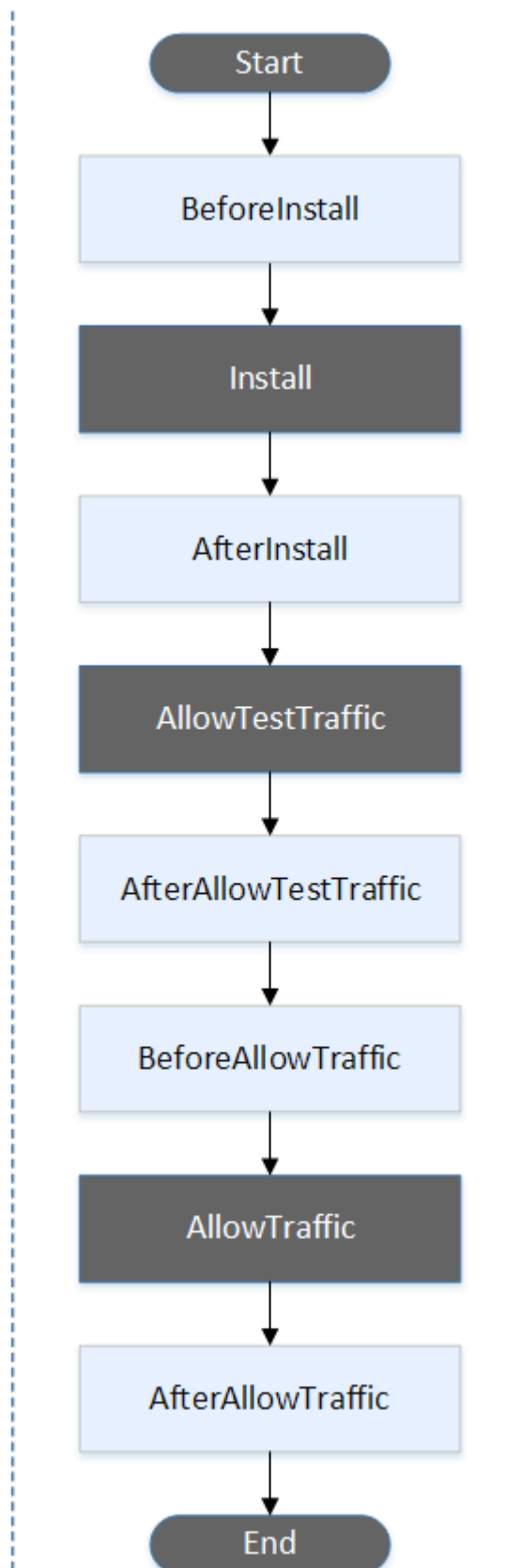
Un hook AWS Lambda est une fonction Lambda spécifiée par une chaîne sur une nouvelle ligne après le nom de l'événement du cycle de vie. Chaque hook est exécuté une fois par déploiement. Vous trouverez ci-dessous les descriptions des événements du cycle de vie au cours desquels vous pouvez exécuter un hook lors d'un déploiement d'Amazon ECS.

- **BeforeInstall**— À utiliser pour exécuter des tâches avant la création de l'ensemble de tâches de remplacement. Un groupe cible est associé à l'ensemble de tâches d'origine. Si un écouteur de test facultatif est spécifié, il est associé à l'ensemble de tâches d'origine. Aucune restauration n'est possible à ce stade.
- **AfterInstall**— À utiliser pour exécuter des tâches une fois que l'ensemble de tâches de remplacement a été créé et que l'un des groupes cibles y est associé. Si un écouteur de test facultatif est spécifié, il est associé à l'ensemble de tâches d'origine. Les résultats de la fonction hook de cet événement du cycle de vie peuvent entraîner une restauration.
- **AfterAllowTestTraffic**— À utiliser pour exécuter des tâches une fois que le récepteur de test a acheminé le trafic vers l'ensemble de tâches de remplacement. À ce stade, les résultats de la fonction hook peuvent entraîner une restauration.
- **BeforeAllowTraffic**— À utiliser pour exécuter des tâches une fois que le deuxième groupe cible est associé à l'ensemble de tâches de remplacement, mais avant que le trafic ne soit transféré vers l'ensemble de tâches de remplacement. Les résultats de la fonction hook de cet événement du cycle de vie peuvent entraîner une restauration.
- **AfterAllowTraffic**— À utiliser pour exécuter des tâches une fois que le deuxième groupe cible a acheminé le trafic vers l'ensemble de tâches de remplacement. Les résultats de la fonction hook de cet événement du cycle de vie peuvent entraîner une restauration.

Pour plus d'informations, consultez [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#) et [Tutoriel : Déployer un service Amazon ECS avec un test de validation](#).

Exécutez l'ordre des hooks dans un déploiement Amazon ECS.

Dans un déploiement Amazon ECS, les hooks d'événements s'exécutent dans l'ordre suivant :



 Note

Les événements de début TestTrafficAllowTraffic, d'installation et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme.

Structure de la section « crochets »

Voici des exemples de structure de la section 'hooks'.

En utilisant YAML :

Hooks:

- BeforeInstall: "*BeforeInstallHookFunctionName*"
- AfterInstall: "*AfterInstallHookFunctionName*"
- AfterAllowTestTraffic: "*AfterAllowTestTrafficHookFunctionName*"
- BeforeAllowTraffic: "*BeforeAllowTrafficHookFunctionName*"
- AfterAllowTraffic: "*AfterAllowTrafficHookFunctionName*"

En utilisant JSON :

```
"Hooks": [
 {
 "BeforeInstall": "BeforeInstallHookFunctionName"
 },
 {
 "AfterInstall": "AfterInstallHookFunctionName"
 },
 {
 "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
 },
 {
 "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
 },
 {
 "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
 }
]
```

## Exemple de fonction « hooks » Lambda

Utilisez 'hooks' cette section pour spécifier une fonction Lambda qui CodeDeploy peut être appelée pour valider un déploiement Amazon ECS. Vous pouvez utiliser la même fonction ou une autre pour les événements du cycle de vie du AfterAllowTraffic déploiement BeforeInstall AfterInstall AfterAllowTestTrafficBeforeAllowTraffic,,,. Une fois les tests de validation terminés, la AfterAllowTraffic fonction Lambda rappelle CodeDeploy et fournit un résultat de Succeeded ou. Failed

### Important

Le déploiement est considéré comme ayant échoué s'il n' CodeDeploy est pas notifié par la fonction de validation Lambda dans un délai d'une heure.

Avant d'appeler une fonction de hook Lambda, le serveur doit être informé de l'ID de déploiement et de l'ID d'exécution du hook d'événement du cycle de vie à l'aide putLifecycleEventHookExecutionStatus de la commande.

Voici un exemple de fonction de crochet Lambda écrite dans le fichier Node.js.

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
 //Read the DeploymentId from the event payload.
 var deploymentId = event.DeploymentId;

 //Read the LifecycleEventHookExecutionId from the event payload
 var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

 /*
 Enter validation tests here.
 */

 // Prepare the validation test results with the deploymentId and
 // the lifecycleEventHookExecutionId for CodeDeploy.
 var params = {
 deploymentId: deploymentId,
```

```
 lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
 status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
 };

 // Pass CodeDeploy the prepared validation test results.
 codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
 if (err) {
 // Validation failed.
 callback('Validation test failed');
 } else {
 // Validation succeeded.
 callback(null, 'Validation test succeeded');
 }
 });
};
```

## AppSpec section « hooks » pour un déploiement AWS Lambda

### Rubriques

- [Liste des hooks d'événements du cycle de vie pour un déploiement AWS Lambda](#)
- [Ordre d'exécution des hooks dans le déploiement d'une version de fonction Lambda](#)
- [Structure de la section « crochets »](#)
- [Exemple de fonction « hooks » Lambda](#)

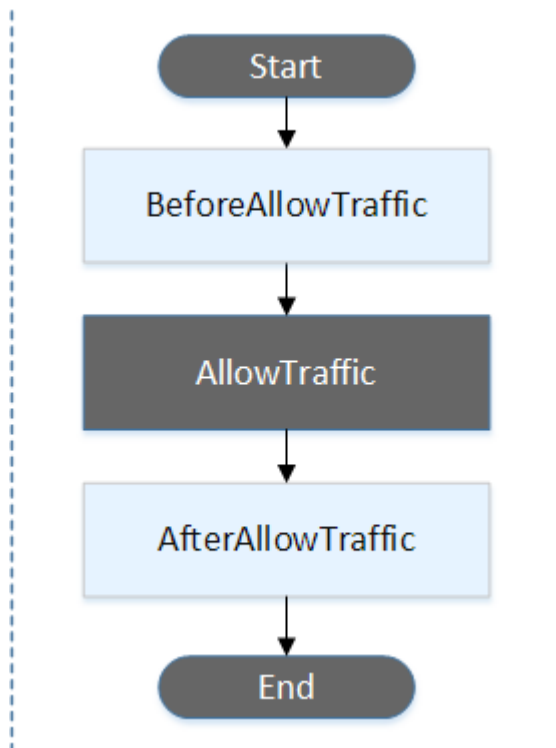
### Liste des hooks d'événements du cycle de vie pour un déploiement AWS Lambda

Un hook AWS Lambda est une fonction Lambda spécifiée par une chaîne sur une nouvelle ligne après le nom de l'événement du cycle de vie. Chaque hook est exécuté une fois par déploiement. Voici les descriptions des crochets pouvant être utilisés dans votre AppSpec fichier.

- **BeforeAllowTraffic**— À utiliser pour exécuter des tâches avant que le trafic ne soit transféré vers la version de la fonction Lambda déployée.
- **AfterAllowTraffic**— À utiliser pour exécuter des tâches une fois que tout le trafic est transféré vers la version de la fonction Lambda déployée.

### Ordre d'exécution des hooks dans le déploiement d'une version de fonction Lambda

Dans le cadre d'un déploiement de la version d'une fonction Lambda sans serveur, les hooks d'événements s'exécutent dans l'ordre suivant :

**Note**

Les événements de début et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme. AllowTraffic

Structure de la section « crochets »

Voici des exemples de structure de la section « hooks ».

En utilisant YAML :

```
hooks:
 - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
 - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

En utilisant JSON :

```
"hooks": [{
 "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
},
{
```



```
"AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

## Exemple de fonction « hooks » Lambda

Utilisez la section « hooks » pour spécifier une fonction Lambda qui CodeDeploy peut être appelée pour valider un déploiement Lambda. Vous pouvez utiliser la même fonction ou une fonction différente pour les événements du cycle de vie `BeforeAllowTraffic` et de `AfterAllowTraffic` déploiement. Une fois les tests de validation terminés, la fonction de validation Lambda rappelle CodeDeploy et fournit un résultat de `Succeeded` ou `Failed`.

### Important

Le déploiement est considéré comme ayant échoué s'il n' CodeDeploy est pas notifié par la fonction de validation Lambda dans un délai d'une heure.

Avant d'appeler une fonction de hook Lambda, le serveur doit être informé de l'ID de déploiement et de l'ID d'exécution du hook d'événement du cycle de vie à l'aide `putLifecycleEventHookExecutionStatus` de la commande.

Voici un exemple de fonction de crochet Lambda écrite dans le fichier `Node.js`.

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
 //Read the DeploymentId from the event payload.
 var deploymentId = event.DeploymentId;

 //Read the LifecycleEventHookExecutionId from the event payload
 var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

 /*
 Enter validation tests here.
 */

 // Prepare the validation test results with the deploymentId and
 // the lifecycleEventHookExecutionId for CodeDeploy.
```

```
var params = {
 deploymentId: deploymentId,
 lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
 status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
};

// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
 if (err) {
 // Validation failed.
 callback('Validation test failed');
 } else {
 // Validation succeeded.
 callback(null, 'Validation test succeeded');
 }
});
};
```

AppSpec section « hooks » pour un déploiement EC2/sur site

## Rubriques

- [Liste des hooks d'événements liés au cycle de vie](#)
- [Disponibilité du carnet d'événements du cycle de vie](#)
- [Exécuter l'ordre des hooks dans un déploiement](#)
- [Structure de la section « crochets »](#)
- [Référencement de fichiers dans vos scripts hook](#)
- [Disponibilité des variables d'environnement pour les hooks](#)
- [Exemple de crochets](#)

## Liste des hooks d'événements liés au cycle de vie

Un hook de déploiement EC2/sur site est exécuté une fois par déploiement sur une instance. Vous pouvez spécifier un ou plusieurs scripts dans un hook. Chaque hook pour un événement de cycle de vie est spécifié avec une chaîne sur une ligne distincte. Voici les descriptions des crochets pouvant être utilisés dans votre AppSpec fichier.

Pour plus d'informations sur les hooks d'événement de cycle de vie qui sont valides pour des types de déploiement et de restauration spécifiques, consultez [Disponibilité du carnet d'événements du cycle de vie](#).

- **ApplicationStop**— Cet événement du cycle de vie du déploiement se produit avant même le téléchargement de la révision de l'application. Vous pouvez spécifier des scripts pour que cet événement arrête l'application sans heurt ou supprime les packages actuellement installés en vue d'un déploiement. Le AppSpec fichier et les scripts utilisés pour cet événement du cycle de vie du déploiement proviennent de la version précédente de l'application déployée avec succès.

 Note

Aucun AppSpec fichier n'existe sur une instance avant que vous ne le déployiez. Pour cette raison, le hook `ApplicationStop` ne s'exécute pas la première fois que vous effectuez un déploiement sur cette instance. Vous pouvez utiliser le hook `ApplicationStop` la deuxième fois que vous effectuez un déploiement sur une instance.

Pour déterminer l'emplacement de la dernière révision d'application déployée avec succès, l' `CodeDeploy` agent recherche l'emplacement indiqué dans le `deployment-group-id_last_successful_install` fichier. Ce fichier se trouve dans :

`/opt/codedeploy-agent/deployment-root/deployment-instructions`dossier sur les instances Amazon Linux, Ubuntu Server et RHEL Amazon EC2.

`C:\ProgramData\Amazon\CodeDeploy\deployment-instructions`dossier sur les instances Amazon EC2 de Windows Server.

Pour dépanner un déploiement qui échoue au cours de l'événement de cycle de vie de déploiement `ApplicationStop`, consultez [Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic](#).

- **DownloadBundle**— Au cours de cet événement du cycle de vie du déploiement, l' `CodeDeploy` agent copie les fichiers de révision de l'application dans un emplacement temporaire :

`/opt/codedeploy-agent/deployment-root/deployment-group-id/deployment-id/deployment-archived`dossier sur les instances Amazon Linux, Ubuntu Server et RHEL Amazon EC2.

`C:\ProgramData\Amazon\CodeDeploy\deployment-group-id\deployment-id\deployment-archived`dossier sur les instances Amazon EC2 de Windows Server.

Cet événement est réservé à l' CodeDeploy agent et ne peut pas être utilisé pour exécuter des scripts.

Pour dépanner un déploiement qui échoue au cours de l'événement de cycle de vie de déploiement `DownloadBundle`, consultez [Résolution des problèmes liés à un échec du cycle de vie d'un `DownloadBundle` déploiement avec `UnknownError` : non ouvert pour lecture](#).

- `BeforeInstall`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour les tâches de préinstallation, telles que le déchiffrement de fichiers et la création d'une sauvegarde de la version actuelle.
- `Install`— Au cours de cet événement du cycle de vie du déploiement, l' CodeDeploy agent copie les fichiers de révision de l'emplacement temporaire vers le dossier de destination final. Cet événement est réservé à l' CodeDeploy agent et ne peut pas être utilisé pour exécuter des scripts.
- `AfterInstall`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour des tâches telles que la configuration de votre application ou la modification des autorisations des fichiers.
- `ApplicationStart`— Vous utilisez généralement cet événement du cycle de vie du déploiement pour redémarrer les services qui ont été interrompus pendant le déploiement `ApplicationStop`.
- `ValidateService`— Il s'agit du dernier événement du cycle de vie du déploiement. Il permet de vérifier que le déploiement a réussi.
- `BeforeBlockTraffic`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour exécuter des tâches sur des instances avant qu'elles ne soient désenregistrées d'un équilibreur de charge.

Pour dépanner un déploiement qui échoue au cours de l'événement de cycle de vie de déploiement `BeforeBlockTraffic`, consultez [Résolution des problèmes liés à un échec `ApplicationStop` ou à un événement lié `AfterBlockTraffic` au cycle de vie du déploiement `BeforeBlockTraffic`](#).

- `BlockTraffic`— Au cours de cet événement du cycle de vie du déploiement, le trafic Internet est empêché d'accéder aux instances qui servent actuellement du trafic. Cet événement est réservé à l' CodeDeploy agent et ne peut pas être utilisé pour exécuter des scripts.
- `AfterBlockTraffic`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour exécuter des tâches sur des instances après leur désenregistrement de leur équilibreur de charge respectif.

Pour dépanner un déploiement qui échoue au cours de l'événement de cycle de vie de déploiement `AfterBlockTraffic`, consultez [Résolution des problèmes liés à un échec `ApplicationStop` ou à un événement lié `AfterBlockTraffic` au cycle de vie du déploiement `BeforeBlockTraffic`](#).

- `BeforeAllowTraffic`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour exécuter des tâches sur des instances avant qu'elles ne soient enregistrées auprès d'un équilibreur de charge.
- `AllowTraffic`— Au cours de cet événement du cycle de vie du déploiement, le trafic Internet est autorisé à accéder aux instances après un déploiement. Cet événement est réservé à l'CodeDeploy agent et ne peut pas être utilisé pour exécuter des scripts.
- `AfterAllowTraffic`— Vous pouvez utiliser cet événement du cycle de vie du déploiement pour exécuter des tâches sur des instances après leur enregistrement auprès d'un équilibreur de charge.

## Disponibilité du carnet d'événements du cycle de vie

Le tableau suivant répertorie les hooks d'événement de cycle de vie disponibles pour chaque scénario de déploiement et de restauration.

| Nom de l'événement de cycle de vie      | Déploiement de lancement d'Auto Scaling <sup>1</sup> | Arrêt du déploiement d'Auto Scaling <sup>1</sup> | Déploiement sur place <sup>2</sup> | Déploiement bleu/vert : Instances d'origine | Déploiement bleu/vert : Instances de remplacement | Restauration de déploiement bleu/vert : Instances d'origine | Restauration de déploiement bleu/vert : Instances de remplacement |
|-----------------------------------------|------------------------------------------------------|--------------------------------------------------|------------------------------------|---------------------------------------------|---------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------|
| <code>ApplicationStop</code>            | ✓                                                    | ✓                                                | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| <code>DownloadBundle<sup>3</sup></code> | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |

| Nom de l'événement de cycle de vie | Déploiement de lancement d'Auto Scaling <sup>1</sup> | Arrêt du déploiement d'Auto Scaling <sup>1</sup> | Déploiement sur place <sup>2</sup> | Déploiement bleu/vert : Instances d'origine | Déploiement bleu/vert : Instances de remplacement | Restauration de déploiement bleu/vert : Instances d'origine | Restauration de déploiement bleu/vert : Instances de remplacement |
|------------------------------------|------------------------------------------------------|--------------------------------------------------|------------------------------------|---------------------------------------------|---------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------|
| BeforeInstall                      | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| Installez <sup>3</sup>             | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| AfterInstall                       | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| ApplicationStart                   | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| ValidateService                    | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 |                                                             |                                                                   |
| BeforeBlockTraffic                 |                                                      | ✓                                                | ✓                                  | ✓                                           |                                                   |                                                             | ✓                                                                 |
| BlockTraffic <sup>3</sup>          |                                                      | ✓                                                | ✓                                  | ✓                                           |                                                   |                                                             | ✓                                                                 |
| AfterBlockTraffic                  |                                                      | ✓                                                | ✓                                  | ✓                                           |                                                   |                                                             | ✓                                                                 |
| BeforeAllowTraffic                 | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 | ✓                                                           |                                                                   |
| AllowTraffic <sup>3</sup>          | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 | ✓                                                           |                                                                   |

| Nom de l'événement de cycle de vie | Déploiement de lancement d'Auto Scaling <sup>1</sup> | Arrêt du déploiement d'Auto Scaling <sup>1</sup> | Déploiement sur place <sup>2</sup> | Déploiement bleu/vert : Instances d'origine | Déploiement bleu/vert : Instances de remplacement | Restauration de déploiement bleu/vert : Instances d'origine | Restauration de déploiement bleu/vert : Instances de remplacement |
|------------------------------------|------------------------------------------------------|--------------------------------------------------|------------------------------------|---------------------------------------------|---------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------|
| AfterAllowTraffic                  | ✓                                                    |                                                  | ✓                                  |                                             | ✓                                                 | ✓                                                           |                                                                   |

<sup>1</sup> Pour plus d'informations sur les déploiements d'Amazon EC2 Auto Scaling, consultez. [Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy](#)

<sup>2</sup> S'applique également à l'annulation d'un déploiement sur place.

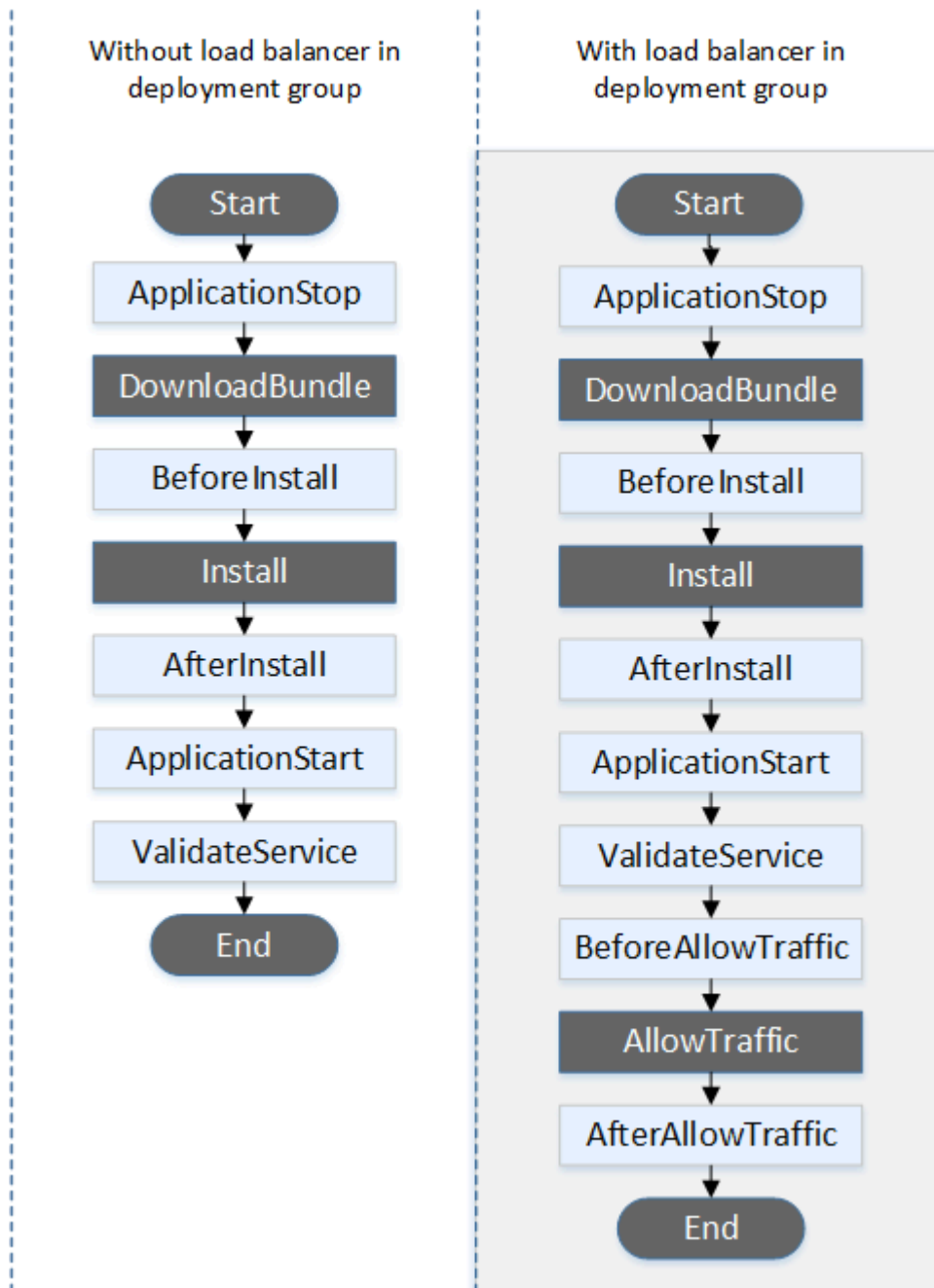
<sup>3</sup> Réservé aux CodeDeploy opérations. Ne peut pas être utilisé pour exécuter des scripts.

## Exécuter l'ordre des hooks dans un déploiement

### Déploiements de lancement d'Auto Scaling

Lors d'un déploiement de lancement d'Auto Scaling, CodeDeploy exécute les hooks d'événements dans l'ordre suivant.

Pour plus d'informations sur les déploiements de lancement d'Auto Scaling, consultez [Comment fonctionne Amazon EC2 Auto Scaling avec CodeDeploy](#).

**Note**

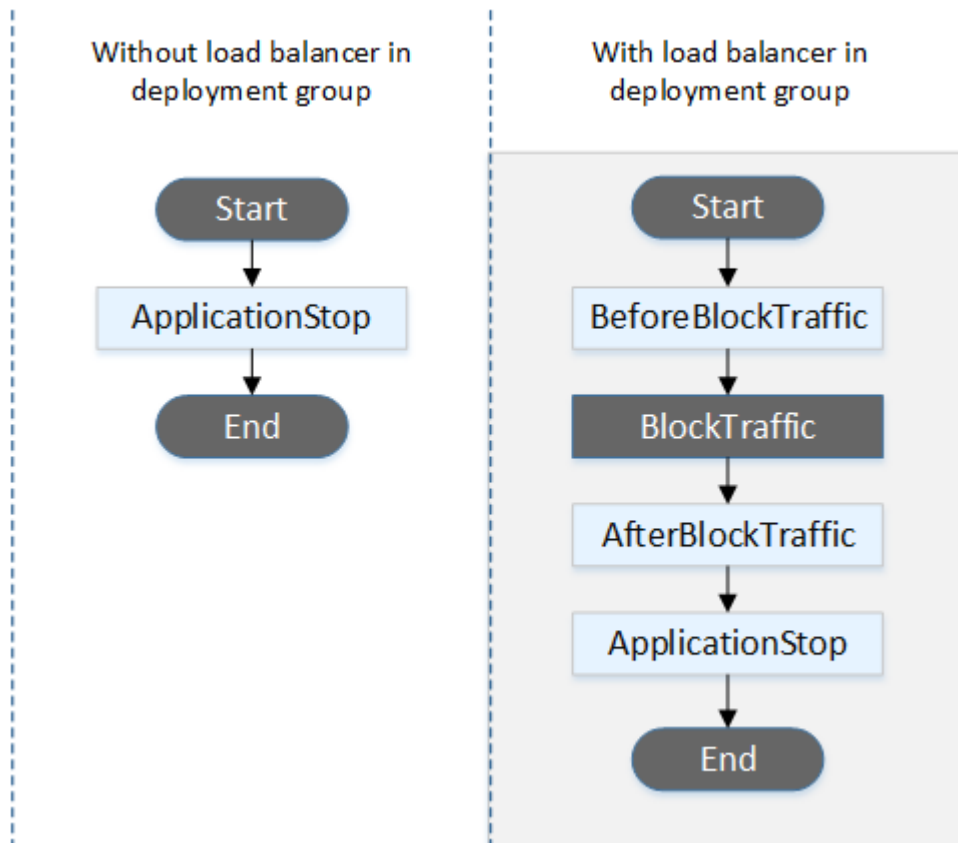
Les événements de début DownloadBundle, d'installation et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme. AllowTraffic Vous pouvez toutefois modifier la 'files' section du AppSpec fichier pour spécifier ce qui est installé lors de l'événement d'installation.



## Déploiements de terminaison Auto Scaling

Lors d'un déploiement de terminaison d'Auto Scaling, CodeDeploy exécute les hooks d'événements dans l'ordre suivant.

Pour plus d'informations sur les déploiements de terminaison Auto Scaling, consultez [Permettre les déploiements de terminaison lors d'événements de scale-in d'Auto Scaling](#).



### Note

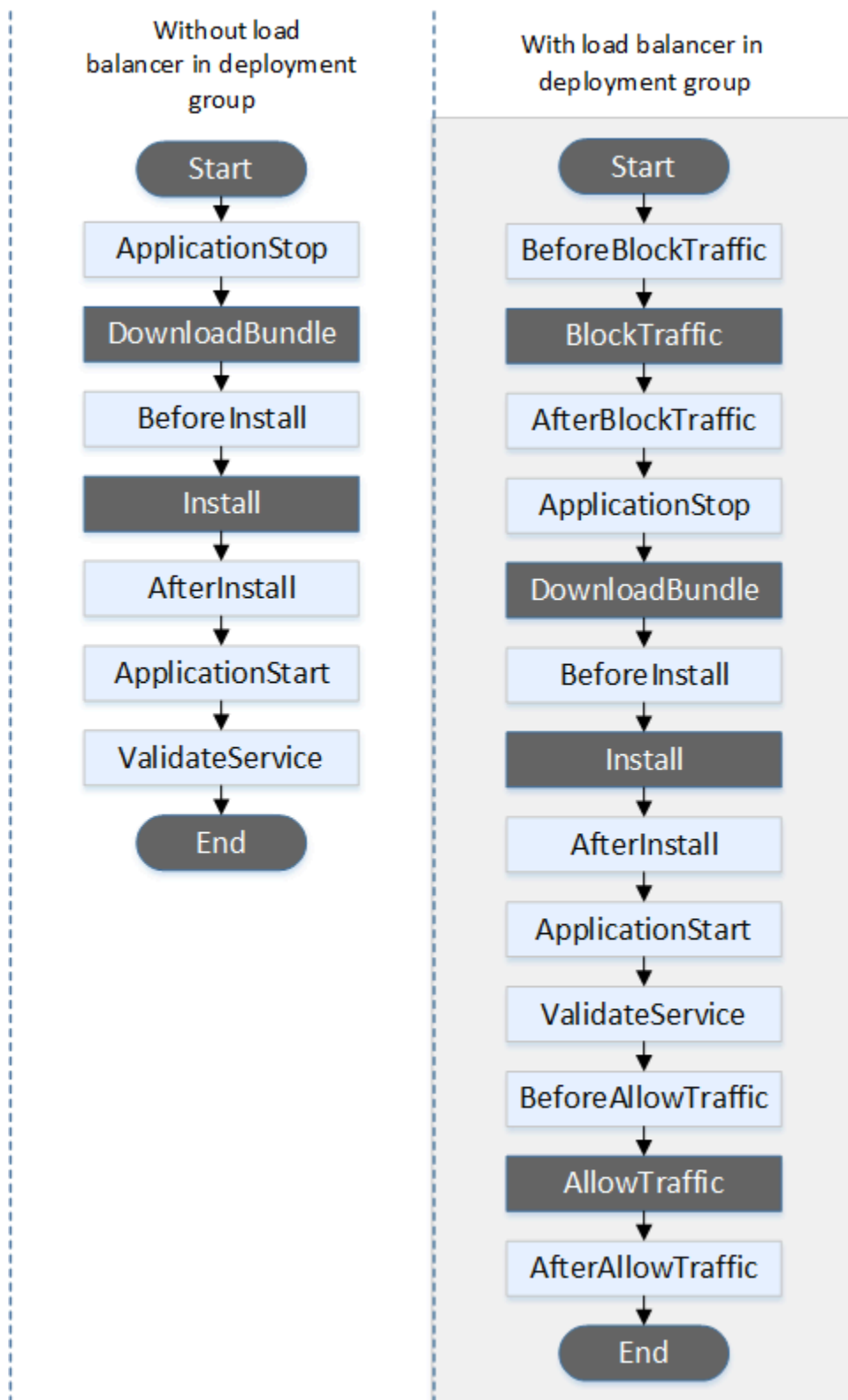
Les événements de début et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme. BlockTraffic

## Déploiements sur place

Dans un déploiement sur place, y compris la restauration d'un déploiement sur place, les hooks d'événement sont exécutés dans l'ordre suivant :

 **Note**

Pour les déploiements sur place, les six hooks liés au blocage et à l'autorisation du trafic s'appliquent uniquement si vous spécifiez un Classic Load Balancer, un Application Load Balancer ou un Network Load Balancer d'Elastic Load Balancing dans le groupe de déploiement.

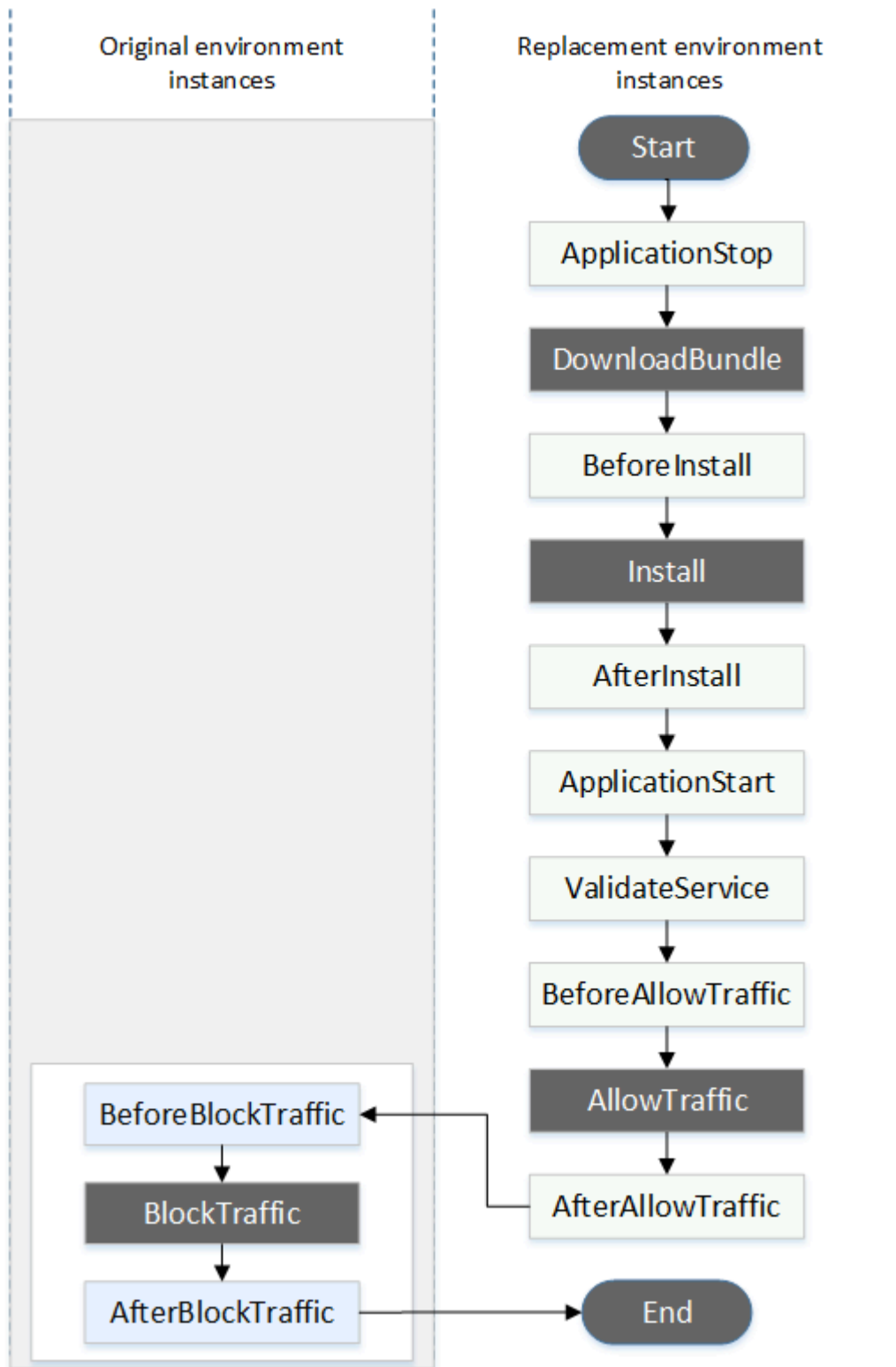


 Note

Les événements de début DownloadBundle, d'installation et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme. Vous pouvez toutefois modifier la 'files' section du AppSpec fichier pour spécifier ce qui est installé lors de l'événement d'installation.

## Déploiements bleu/vert

Lors d'un blue/green déploiement, les hooks d'événements sont exécutés dans l'ordre suivant :



 Note

Les événements de début DownloadBundle, d'installation et de fin du déploiement ne peuvent pas être scriptés, c'est pourquoi ils apparaissent en gris dans ce diagramme. BlockTrafficAllowTraffic Cependant, vous pouvez modifier la section « fichiers » du AppSpec fichier pour spécifier ce qui est installé lors de l'événement d'installation.

## Structure de la section « crochets »

La section ' hooks ' a la structure suivante :

```
hooks :
 deployment-lifecycle-event-name :
 - location: script-location
 timeout: timeout-in-seconds
 runas: user-name
```

Vous pouvez inclure les éléments suivants dans une entrée hook après le nom de l'événement de cycle de vie de déploiement :

### location

Obligatoire. Emplacement dans le groupe du fichier de script pour la révision. L'emplacement des scripts que vous spécifiez dans hooks cette section est relatif à la racine du bundle de révision de l'application. Pour de plus amples informations, veuillez consulter [Planifier une révision pour CodeDeploy](#).

### timeout

Facultatif. Nombre de secondes permettant au script de s'exécuter avant qu'il soit considéré comme ayant échoué. La valeur par défaut est de 3600 secondes (1 heure).

 Note

La durée de 3600 secondes (1 heure) représente le laps de temps maximal autorisé pour l'exécution du script pour chaque événement du cycle de vie de déploiement. Si des scripts dépassent cette limite, le déploiement s'arrête et le déploiement vers l'instance échoue. Veillez à ce que le nombre total de secondes spécifié dans timeout pour tous les scripts de chaque événement du cycle de vie de déploiement ne dépasse pas cette limite.

## runas

Facultatif. Utilisateur dont vous empruntez l'identité lorsque vous exécutez le script. Par défaut, il s'agit de l' CodeDeploy agent exécuté sur l'instance. CodeDeploy ne stocke pas les mots de passe, de sorte que l'utilisateur ne peut pas être usurpé si l'utilisateur runas a besoin d'un mot de passe. Cet élément s'applique uniquement aux instances Amazon Linux et Ubuntu Server.

## Référencement de fichiers dans vos scripts hook

Si vous associez un script à un événement CodeDeploy du cycle de vie comme décrit dans [AppSpec section « crochets »](#), et que vous souhaitez référencer un fichier (par exemple, `helper.sh`) dans votre script, vous devez spécifier `helper.sh` en utilisant :

- (Recommandé) Un chemin absolu. Consultez [Utilisation de chemins absolus](#).
- Un chemin relatif. Consultez [Utilisation de chemins relatifs](#).

## Utilisation de chemins absolus

Pour référencer un fichier à l'aide de son chemin absolu, vous pouvez soit :

- Spécifiez le chemin absolu dans la `files` section du AppSpec fichier, dans la `destination` propriété. Spécifiez ensuite le même chemin absolu dans votre script hook. Pour de plus amples informations, veuillez consulter [AppSpec section « fichiers » \(déploiements EC2/sur site uniquement\)](#).
- Spécifiez un chemin absolu dynamique dans votre script hook. Pour plus d'informations, consultez la section [Emplacement de l'archive de déploiement](#).

## Emplacement de l'archive de déploiement

Au cours de l'événement du [DownloadBundle](#) cycle de vie, l' CodeDeploy agent extrait la [révision](#) pour le déploiement dans un répertoire au format suivant :

*root-directory/deployment-group-id/deployment-id/deployment-archive*

La *root-directory* partie du chemin est toujours définie sur la valeur par défaut indiquée dans le tableau suivant ou est contrôlée par le paramètre `:root_dir` de configuration. Pour plus d'informations sur les paramètres de configuration, consultez [CodeDeploy référence de configuration de l'agent](#).

| Plateforme d'agents                          | Répertoire racine par défaut          |  |
|----------------------------------------------|---------------------------------------|--|
| Linux : toutes les distributions rpm         | /opt/codedeploy-agent/deployment-root |  |
| Ubuntu Server — toutes les distributions deb | /opt/codedeploy-agent/deployment-root |  |
| Windows Server                               | %ProgramData%\Amazon\CodeDeploy       |  |

À partir de vos scripts hook, vous pouvez accéder à l'archive de déploiement actuelle en utilisant le chemin du répertoire racine `DEPLOYMENT_ID` et les variables d'environnement `DEPLOYMENT_GROUP_ID`. Pour plus d'informations sur les variables que vous pouvez utiliser, consultez [Disponibilité des variables d'environnement pour les hooks](#).

Par exemple, voici comment accéder à un `data.json` fichier qui se trouve à la racine de votre révision sous Linux :

```
#!/bin/bash

rootDirectory="/opt/codedeploy-agent/deployment-root" # note: this will be different if
you # customize the :root_dir
configuration
dataFile="$rootDirectory/$DEPLOYMENT_GROUP_ID/$DEPLOYMENT_ID/deployment-archive/
data.json"
data=$(cat dataFile)
```

Autre exemple, voici comment accéder à un `data.json` fichier qui se trouve à la racine de votre révision à l'aide de Powershell sous Windows :

```
$rootDirectory="$env:ProgramData\Amazon\CodeDeploy" # note: this will be different if
you # customize the :root_dir
configuration
$dataFile="$rootDirectory\$env:DEPLOYMENT_GROUP_ID\$env:DEPLOYMENT_ID\deployment-
archive\data.json"
$data=(Get-Content $dataFile)
```



## Utilisation de chemins relatifs

Pour référencer un fichier à l'aide de son chemin relatif, vous devez connaître le répertoire de travail de l' CodeDeploy agent. Les chemins de fichiers sont relatifs à ce répertoire.

Le tableau suivant indique le répertoire de travail pour chaque plate-forme prise en charge par l' CodeDeploy agent.

| Plateforme d'agents                             | Méthode de gestion des processus        | Répertoire de travail pour les scripts d'événements du cycle de vie |
|-------------------------------------------------|-----------------------------------------|---------------------------------------------------------------------|
| Linux : toutes les distributions rpm            | systemd (par défaut)                    | /                                                                   |
|                                                 | <a href="#">init.d — En savoir plus</a> | /opt/codedeploy-agent                                               |
| Ubuntu Server — toutes les distributions Debian | tout                                    | /opt/codedeploy-agent                                               |
| Windows Server                                  | non applicable                          | C:\Windows\System32                                                 |

## Disponibilité des variables d'environnement pour les hooks

Au cours de chaque événement du cycle de vie de déploiement, les scripts de hook peuvent accéder aux variables d'environnement suivantes :

### APPLICATION\_NAME

Le nom de l'application CodeDeploy qui y figure fait partie du déploiement en cours (par exemple,WordPress\_App).

### DEPLOYMENT\_ID

L'ID CodeDeploy a été attribué au déploiement en cours (par exemple,d-AB1CDEF23).

### DEPLOYMENT\_GROUP\_NAME

Le nom du groupe de déploiement CodeDeploy qui fait partie du déploiement en cours (par exemple,WordPress\_DepGroup).

## DEPLOYMENT\_GROUP\_ID

L'ID du groupe de déploiement CodeDeploy qui fait partie du déploiement en cours (par exemple, `b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE`).

## LIFECYCLE\_EVENT

Nom de l'événement de cycle de vie de déploiement actuel (par exemple, `AfterInstall`).

Ces variables d'environnement sont locales à chaque événement de cycle de vie de déploiement.

Des variables d'environnement supplémentaires sont disponibles pour connecter les scripts en fonction de la source du bundle de déploiement :

### Bundle d'Amazon S3

- **BUNDLE\_BUCKET**

Le nom du compartiment Amazon S3 à partir duquel le bundle de déploiement a été téléchargé (par exemple, `my-s3-bucket`).

- **BUNDLE\_KEY**

La clé d'objet du bundle téléchargé dans le compartiment Amazon S3 (par exemple, `WordPress_App.zip`).

- **VERSION DU BUNDLE**

Version de l'objet pour le bundle (par exemple, `3sL4kqtJ1cpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40N18X8gdRQBpUMLUo`). Cette variable n'est définie que si le contrôle de [version des objets](#) est activé dans le compartiment Amazon S3.

- **BUNDLE\_ETAG**

L'étiquette d'objet du bundle (par exemple, `b10a8db164e0754105b7a99be72e3fe5-4`).

### Offre groupée provenant de GitHub

- **BUNDLE\_COMMIT**

Le hachage de SHA256 validation du bundle généré par Git (par exemple, `d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26`).

Le script suivant modifie le port d'écoute sur un serveur Apache HTTP en spécifiant 9090 à la place de 80 si la valeur de `DEPLOYMENT_GROUP_NAME` est égale à `Staging`. Ce script doit être appelé au cours de l'événement de cycle de vie de déploiement `BeforeInstall` :

```
if ["$DEPLOYMENT_GROUP_NAME" == "Staging"]
then
 sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

L'exemple de script suivant modifie le niveau de détail des messages enregistrés dans son journal des erreurs et le fait passer d'avertissement à débogage si la valeur de la variable d'environnement `DEPLOYMENT_GROUP_NAME` est égale à `Staging`. Ce script doit être appelé au cours de l'événement de cycle de vie de déploiement `BeforeInstall` :

```
if ["$DEPLOYMENT_GROUP_NAME" == "Staging"]
then
 sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

L'exemple de script suivant remplace le texte de la page Web spécifiée par un texte qui affiche la valeur de ces variables d'environnement. Ce script doit être appelé au cours de l'événement de cycle de vie de déploiement `AfterInstall` :

```
#!/usr/bin/python

import os

strToSearch("<h2>This application was deployed using CodeDeploy.</h2>")
strToReplace("<h2>This page for "+os.environ['APPLICATION_NAME']+"
application and "+os.environ['DEPLOYMENT_GROUP_NAME']+" deployment group with
"+os.environ['DEPLOYMENT_GROUP_ID']+" deployment group ID was generated by a
"+os.environ['LIFECYCLE_EVENT']+" script during "+os.environ['DEPLOYMENT_ID']+"
deployment.</h2>")

fp=open("/var/www/html/index.html","r")
buffer=fp.read()
fp.close()

fp=open("/var/www/html/index.html","w")
fp.write(buffer.replace(strToSearch,strToReplace))
fp.close()
```

## Exemple de crochets

Voici un exemple d'entrée hooks qui spécifie deux hooks pour l'événement de cycle de vie `AfterInstall` :

```
hooks:
 AfterInstall:
 - location: Scripts/RunResourceTests.sh
 timeout: 180
 - location: Scripts/PostDeploy.sh
 timeout: 180
```

Le script `Scripts/RunResourceTests.sh` s'exécute au cours de la phase `AfterInstall` du processus de déploiement. Le déploiement échoue si l'exécution du script dure plus de 180 secondes (3 minutes).

L'emplacement des scripts que vous spécifiez dans la section « hooks » est relatif à la racine du groupe de révision d'application. Dans l'exemple précédent, un fichier nommé `RunResourceTests.sh` se trouve dans un répertoire nommé `Scripts`. Le répertoire `Scripts` est à la racine du groupe. Pour de plus amples informations, veuillez consulter [Planifier une révision pour CodeDeploy](#).

## AppSpec Exemple de fichier

Cette rubrique fournit des AppSpec fichiers d'exemple pour un déploiement AWS Lambda et EC2/on-premises.

### Rubriques

- [AppSpec Exemple de fichier pour un déploiement Amazon ECS](#)
- [AppSpec Exemple de fichier pour un déploiement AWS Lambda](#)
- [AppSpec Exemple de fichier pour un déploiement EC2/sur site](#)

## AppSpec Exemple de fichier pour un déploiement Amazon ECS

Voici un exemple de AppSpec fichier écrit en YAML pour déployer un service Amazon ECS.

```
version: 0.0
Resources:
 - TargetService:
```

```

 Type: AWS::ECS::Service
 Properties:
 TaskDefinition: "arn:aws:ecs:us-east-1:111222333444:task-definition/my-task-
definition-family-name:1"
 LoadBalancerInfo:
 ContainerName: "SampleApplicationName"
 ContainerPort: 80
Optional properties
 PlatformVersion: "LATEST"
 NetworkConfiguration:
 AwsVpcConfiguration:
 Subnets: ["subnet-1234abcd", "subnet-5678abcd"]
 SecurityGroups: ["sg-12345678"]
 AssignPublicIp: "ENABLED"
 CapacityProviderStrategy:
 - Base: 1
 CapacityProvider: "FARGATE_SPOT"
 Weight: 2
 - Base: 0
 CapacityProvider: "FARGATE"
 Weight: 1
 Hooks:
 - BeforeInstall: "LambdaFunctionToValidateBeforeInstall"
 - AfterInstall: "LambdaFunctionToValidateAfterInstall"
 - AfterAllowTestTraffic: "LambdaFunctionToValidateAfterTestTrafficStarts"
 - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeAllowingProductionTraffic"
 - AfterAllowTraffic: "LambdaFunctionToValidateAfterAllowingProductionTraffic"

```

Voici la version JSON de l'exemple précédent.

```

{
 "version": 0.0,
 "Resources": [
 {
 "TargetService": {
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "arn:aws:ecs:us-east-1:111222333444:task-
definition/my-task-definition-family-name:1",
 "LoadBalancerInfo": {
 "ContainerName": "SampleApplicationName",
 "ContainerPort": 80
 },

```

```

 "PlatformVersion": "LATEST",
 "NetworkConfiguration": {
 "AwsVpcConfiguration": {
 "Subnets": [
 "subnet-1234abcd",
 "subnet-5678abcd"
],
 "SecurityGroups": [
 "sg-12345678"
],
 "AssignPublicIp": "ENABLED"
 }
 },
 "CapacityProviderStrategy": [
 {
 "Base" : 1,
 "CapacityProvider" : "FARGATE_SPOT",
 "Weight" : 2
 },
 {
 "Base" : 0,
 "CapacityProvider" : "FARGATE",
 "Weight" : 1
 }
]
 }
},
 "Hooks": [
 {
 "BeforeInstall": "LambdaFunctionToValidateBeforeInstall"
 },
 {
 "AfterInstall": "LambdaFunctionToValidateAfterInstall"
 },
 {
 "AfterAllowTestTraffic": "LambdaFunctionToValidateAfterTestTrafficStarts"
 },
 {
 "BeforeAllowTraffic":
"LambdaFunctionToValidateBeforeAllowingProductionTraffic"
 },
 {

```

```
 "AfterAllowTraffic":
 "LambdaFunctionToValidateAfterAllowingProductionTraffic"
 }
]
}
```

Voici la séquence d'événements durant le déploiement :

1. Avant que l'application Amazon ECS mise à jour ne soit installée sur l'ensemble de tâches de remplacement, la fonction Lambda appelée `s'LambdaFunctionToValidateBeforeInstall` exécute.
2. Une fois que l'application Amazon ECS mise à jour est installée sur l'ensemble de tâches de remplacement, mais avant qu'elle ne reçoive le moindre trafic, la fonction Lambda appelée `s'LambdaFunctionToValidateAfterInstall` exécute.
3. Une fois que l'application Amazon ECS associée à l'ensemble de tâches de remplacement commence à recevoir du trafic en provenance de l'écouteur de test, la fonction `LambdaFunctionToValidateAfterTestTrafficStarts` Lambda appelée s'exécute. Cette fonction exécute fréquemment des tests de validation pour déterminer si le déploiement se poursuit. Si aucun écouteur de test n'est défini dans votre groupe de déploiement, le hook est ignoré.
4. Une fois tous les tests de validation effectués dans le `AfterAllowTestTraffic` hook et avant que le trafic de production ne soit envoyé à l'application Amazon ECS mise à jour, la fonction Lambda appelée `s'LambdaFunctionToValidateBeforeAllowingProductionTraffic` exécute.
5. Une fois que le trafic de production est envoyé à l'application Amazon ECS mise à jour sur l'ensemble de tâches de remplacement, la fonction Lambda appelée `s'LambdaFunctionToValidateAfterAllowingProductionTraffic` exécute.

Les fonctions Lambda qui s'exécutent pendant n'importe quel hook peuvent effectuer des tests de validation ou collecter des métriques de trafic.

## AppSpec Exemple de fichier pour un déploiement AWS Lambda

Voici un exemple de AppSpec fichier écrit en YAML pour déployer une version de fonction Lambda.

```
version: 0.0
Resources:
```

```
- myLambdaFunction:
 Type: AWS::Lambda::Function
 Properties:
 Name: "myLambdaFunction"
 Alias: "myLambdaFunctionAlias"
 CurrentVersion: "1"
 TargetVersion: "2"
Hooks:
 - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
 - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Voici la version JSON de l'exemple précédent.

```
{
 "version": 0.0,
 "Resources": [{
 "myLambdaFunction": {
 "Type": "AWS::Lambda::Function",
 "Properties": {
 "Name": "myLambdaFunction",
 "Alias": "myLambdaFunctionAlias",
 "CurrentVersion": "1",
 "TargetVersion": "2"
 }
 }
]},
 "Hooks": [{
 "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
 },
 {
 "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
 }
]
}
```

Voici la séquence d'événements durant le déploiement :

1. Avant de transférer le trafic de la version 1 d'une fonction Lambda appelée myLambdaFunction vers la version 2, exécutez une fonction Lambda appelée LambdaFunctionToValidateBeforeTrafficShift qui valide que le déploiement est prêt à démarrer le transfert de trafic.



2. Si `LambdaFunctionToValidateBeforeTrafficShift` retourne un code de sortie de 0 (succès), commencez à déplacer le trafic vers la version 2 de `myLambdaFunction`. La configuration de ce déploiement détermine le débit auquel le trafic est déplacé.
3. Une fois le transfert du trafic de la version 1 d'une fonction Lambda appelée `myLambdaFunction` à la version 2 terminé, exécutez une fonction Lambda appelée `LambdaFunctionToValidateAfterTrafficShift` qui valide que le déploiement a été effectué avec succès.

## AppSpec Exemple de fichier pour un déploiement EC2/sur site

Voici un exemple de AppSpec fichier pour un déploiement sur place sur une instance Amazon Linux, Ubuntu Server ou RHEL.

### Note

Les déploiements sur des instances Windows Server ne prennent pas en charge cet élément. Si vous effectuez un déploiement sur des instances Windows Server, ne l'incluez pas dans votre AppSpec fichier.

```
version: 0.0
os: linux
files:
 - source: Config/config.txt
 destination: /webapps/Config
 - source: source
 destination: /webapps/myApp
hooks:
 BeforeInstall:
 - location: Scripts/UnzipResourceBundle.sh
 - location: Scripts/UnzipDataBundle.sh
 AfterInstall:
 - location: Scripts/RunResourceTests.sh
 timeout: 180
 ApplicationStart:
 - location: Scripts/RunFunctionalTests.sh
 timeout: 3600
 ValidateService:
 - location: Scripts/MonitorService.sh
 timeout: 3600
```

```
runas: codedeployuser
```

Pour une instance Windows Server, passez `os: linux` à `os: windows`. En outre, vous devez utiliser des noms qualifiés complets pour les chemins destination (par exemple, `c:\temp\webapps\Config` et `c:\temp\webapps\myApp`). N'incluez pas l'élément `runas`.

Voici la séquence d'événements durant le déploiement :

1. Exécutez le script situé dans `Scripts/UnzipResourceBundle.sh`.
2. Si le script précédent a retourné un code de sortie 0 (réussite), exécutez le script situé dans `Scripts/UnzipDataBundle.sh`.
3. Copiez le fichier du chemin d'accès `Config/config.txt` vers le chemin d'accès `/webapps/Config/config.txt`.
4. Copiez de façon récursive tous les fichiers du répertoire source dans le répertoire `/webapps/myApp`.
5. Exécutez le script situé dans `Scripts/RunResourceTests.sh` avec un délai de 180 secondes (3 minutes).
6. Exécutez le script situé dans `Scripts/RunFunctionalTests.sh` avec un délai de 3600 secondes (1 heure).
7. Exécutez le script situé dans `Scripts/MonitorService.sh` en tant qu'utilisateur `codedeploy` avec un délai de 3600 secondes (1 heure).

## AppSpec Espacement des fichiers

Voici le format correct pour l'espacement des AppSpec fichiers. Les chiffres entre crochets indiquent le nombre d'espaces qui doivent figurer entre les éléments. Par exemple, `[4]` cela signifie insérer quatre espaces entre les éléments. CodeDeploy génère une erreur qui peut être difficile à déboguer si les emplacements et le nombre d'espaces dans un AppSpec fichier ne sont pas corrects.

```
version:[1]version-number
os:[1]operating-system-name
files:
[2]-[1]source:[1]source-files-location
[4]destination:[1]destination-files-location
permissions:
[2]-[1]object:[1]object-specification
[4]pattern:[1]pattern-specification
```

```

[4]except:[1]exception-specification
[4]owner:[1]owner-account-name
[4]group:[1]group-name
[4]mode:[1]mode-specification
[4]acls:
[6]-[1]acls-specification
[4]context:
[6]user:[1]user-specification
[6]type:[1]type-specification
[6]range:[1]range-specification
[4]type:
[6]-[1]object-type
hooks:
[2]deployment-lifecycle-event-name:
[4]-[1]location:[1]script-location
[6]timeout:[1]timeout-in-seconds
[6]runas:[1]user-name

```

Voici un exemple de AppSpec fichier correctement espacé :

```

version: 0.0
os: linux
files:
 - source: /
 destination: /var/www/html/WordPress
hooks:
 BeforeInstall:
 - location: scripts/install_dependencies.sh
 timeout: 300
 runas: root
 AfterInstall:
 - location: scripts/change_permissions.sh
 timeout: 300
 runas: root
 ApplicationStart:
 - location: scripts/start_server.sh
 - location: scripts/create_test_db.sh
 timeout: 300
 runas: root
 ApplicationStop:
 - location: scripts/stop_server.sh
 timeout: 300
 runas: root

```

Pour plus d'informations sur l'espacement, consultez la spécification [YAML](#).

## Validez votre AppSpec fichier et son emplacement

### Syntaxe du fichier

Vous pouvez utiliser le script AppSpec Assistant AWS fourni pour valider le contenu d'un AppSpec fichier. Vous pouvez trouver le script ainsi que les modèles de AppSpec fichiers sur [GitHub](#).

Vous pouvez également utiliser un outil basé sur un navigateur tel que [YAML lint](#) ou [Online YAML parser](#) pour vérifier votre syntaxe YAML.

### Emplacement du fichier

Pour vérifier que vous avez placé votre AppSpec fichier dans le répertoire racine de la structure de répertoires du contenu source de l'application, exécutez l'une des commandes suivantes :

Sur les instances Linux, macOS ou Unix locales :

```
ls path/to/root/directory/appspec.yml
```

Si le AppSpec fichier ne s'y trouve pas, le message d'erreur « Aucun fichier ou répertoire de ce type » s'affiche.

Sur les instances Windows locales :

```
dir path\to\root\directory\appspec.yml
```

Si le AppSpec fichier ne s'y trouve pas, le message d'erreur « Fichier introuvable » s'affiche.

## CodeDeploy référence de configuration de l'agent

Lorsque l' CodeDeploy agent est installé, un fichier de configuration est placé sur l'instance. Ce fichier de configuration spécifie les chemins de répertoire et les autres paramètres CodeDeploy à utiliser lorsqu'il interagit avec l'instance. Vous pouvez modifier certaines options de configuration dans ce fichier.

Pour les instances Amazon Linux, Ubuntu Server et Red Hat Enterprise Linux (RHEL), le fichier de configuration est nommé `codedeployagent.yml`. Il est placé dans le répertoire `/etc/codedeploy-agent/conf`.

Pour les instances Windows Server, le fichier de configuration est nommé `conf.yml`. Il est placé dans le répertoire `C:\ProgramData\Amazon\CodeDeploy`.

Les paramètres de configuration incluent les éléments suivants :

`:log_aws_wire:`

Définissez cette valeur **true** pour que l'CodeDeploy agent capture les journaux de transfert depuis Amazon S3 et les écrive **codedeploy-agent.wire.log** dans un fichier nommé à l'emplacement indiqué par le paramètre `log_dir` :

 **Warning**

Vous devez définir `:log_aws_wire:` sur `true` uniquement pendant le temps nécessaire à la capture des journaux avec connexion. Le fichier `codedeploy-agent.wire.log` peut atteindre rapidement une très grande taille. La sortie du journal des connexions dans ce fichier peut contenir des informations sensibles, notamment le contenu en texte brut des fichiers transférés vers ou depuis Amazon S3 lorsque ce paramètre était défini sur `true`. Les journaux de connexion contiennent des informations sur toutes les activités Amazon S3 associées au AWS compte lorsque ce paramètre était défini sur `true`, et pas uniquement sur les activités liées aux CodeDeploy déploiements.

Le paramètre par défaut est `false`.

Ce paramètre s'applique à tous les types d'instance. Vous devez ajouter ce paramètre de configuration aux instances Windows Server pour pouvoir l'utiliser.

`:log_dir:`

Le dossier de l'instance dans lequel sont stockés les fichiers journaux relatifs aux opérations de l' CodeDeploy agent.

Le paramètre par défaut est `'/var/log/aws/codedeploy-agent'` pour les instances Amazon Linux, Ubuntu Server et RHEL, ainsi que `C:\ProgramData\Amazon\CodeDeploy\log` pour les instances Windows Server.

`:pid_dir:`

Dossier dans lequel `codedeploy-agent.pid` est stocké.

Ce fichier contient l'identifiant de processus (PID) de l' CodeDeploy agent. Le paramètre par défaut est `'/opt/codedeploy-agent/state/.pid'`.

Ce paramètre s'applique uniquement aux instances Amazon Linux, Ubuntu Server et RHEL.

`:program_name:`

Le nom du programme de l' CodeDeploy agent.

Le paramètre par défaut est `codedeploy-agent`.

Ce paramètre s'applique uniquement aux instances Amazon Linux, Ubuntu Server et RHEL.

`:root_dir:`

Dossier dans lequel les révisions, l'historique des déploiements et les scripts de déploiement associés sont stockés sur l'instance.

Le paramètre par défaut est `/opt/code deploy-agent/deployment-root` pour les instances Amazon Linux, Ubuntu Server et RHEL, ainsi que `C:\ProgramData\Amazon\CodeDeploy` pour les instances Windows Server.

`:verbose:`

Définissez cette valeur `true` pour que l'CodeDeploy agent imprime les fichiers journaux des messages de débogage sur l'instance.

Le paramètre par défaut est `false`.

`:wait_between_runs:`

Intervalle, en secondes, entre l'interrogation de CodeDeploy l'agent et CodeDeploy les déploiements en attente.

Le paramètre par défaut est `1`.

`:on_premises_config_file:`

Pour les instances locales, chemin d'accès à un autre emplacement pour le fichier de configuration nommé `codedeploy.onpremises.yml` (pour Ubuntu Server et RHEL) ou `conf.onpremises.yml` (pour Windows Server).

Par défaut, ces fichiers sont stockés dans `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml` pour Ubuntu Server et RHEL et `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml` pour Windows Server.

Disponible dans la version 1.0.1.686 et les versions ultérieures de l'agent CodeDeploy.

`:proxy_uri:`

(Facultatif) Le proxy HTTP via lequel vous souhaitez que l'agent CodeDeploy se connecte AWS pour vos CodeDeploy opérations. Utilisez un format semblable à `https://user:password@my.proxy:443/path?query`.

Disponible dans la version 1.0.1.824 et les versions ultérieures de l'agent CodeDeploy.



|                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>:max_revisions:</code>       | <p>(Facultatif) Nombre de révisions d'applications pour un groupe de déploiement que vous souhaitez que l'agent CodeDeploy archive. Toutes les révisions qui dépassent le nombre spécifié sont supprimées.</p> <p>Entrez un nombre entier positif quelconque. Si aucune valeur n'est spécifiée, les cinq révisions les plus récentes CodeDeploy seront conservées en plus de la révision actuellement déployée.</p> <p>Pris en charge dans la version 1.0.1.966 et les versions ultérieures de l'agent CodeDeploy.</p> |
| <code>:enable_auth_policy :</code> | <p>(Facultatif) Définissez cette option <code>true</code> si vous souhaitez utiliser l'<a href="#">autorisation IAM</a> pour configurer le contrôle d'accès et limiter les autorisations du rôle ou de l'utilisateur IAM utilisé par l'agent CodeDeploy. Pour <a href="#">Utilisation CodeDeploy avec Amazon Virtual Private Cloud</a>, cette valeur doit être <code>true</code>.</p> <p>Le paramètre par défaut est <code>false</code>.</p>                                                                           |
| <code>:disable_imds_v1 :</code>    | <p>Ce paramètre est disponible avec CodeDeploy l'agent 1.7.0 et versions ultérieures.</p> <p>Définissez cette valeur <code>true</code> pour désactiver la solution de secours en IMDSv1 cas d'erreur IMDSv2. La valeur par défaut est <code>false</code> (active le repli).</p>                                                                                                                                                                                                                                        |

## Rubriques en relation

[Travailler avec l'agent CodeDeploy](#)

[Gestion des opérations des agents CodeDeploy](#)

# CloudFormation modèles de CodeDeploy référence

Cette section présente les CloudFormation ressources, la transformation et le hook conçus pour fonctionner avec les CodeDeploy déploiements. Pour une procédure pas à pas sur la création d'une mise à jour de pile gérée par le CloudFormation hook for CodeDeploy, voir [Créez un blue/green déploiement Amazon ECS via CloudFormation](#)

## Note

CloudFormation les hooks font partie des CloudFormation composants AWS et sont différents des hooks d'événements du CodeDeploy cycle de vie.

Outre les autres méthodes disponibles CodeDeploy, vous pouvez utiliser des CloudFormation modèles pour effectuer les tâches suivantes :

- Création d'applications
- Création de groupes de déploiement et spécification d'une révision cible
- Création de configurations de déploiement
- Créez des instances Amazon EC2.

CloudFormation est un service qui vous aide à modéliser et à configurer vos AWS ressources à l'aide de modèles. Un CloudFormation modèle est un fichier texte dont le format est conforme à la norme JSON. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez, et vous vous CloudFormation occupez de leur provisionnement et de leur configuration.

Pour plus d'informations, voir [Qu'est-ce que c'est AWS CloudFormation ?](#) et [Utilisation de AWS CloudFormation modèles](#) dans le guide de AWS CloudFormation l'utilisateur.

Si vous envisagez d'utiliser des CloudFormation modèles compatibles avec ceux de votre organisation, vous devez, CodeDeploy en tant qu'administrateur, accorder l'accès aux CloudFormation AWS services et aux actions dont ils CloudFormation dépendent. Pour accorder des autorisations permettant de créer des applications, des groupes de déploiement et des configurations de déploiement, ajoutez la politique suivante à l'ensemble d'autorisations des utilisateurs avec lesquels vous travaillerez CloudFormation :

## JSON

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

Pour plus d'informations sur les politiques, consultez les rubriques suivantes :

- Pour consulter la politique qui doit être ajoutée à l'ensemble d'autorisations des utilisateurs qui créeront des instances Amazon EC2, consultez. [Création d'une instance Amazon EC2 pour CodeDeploy \(modèle\)CloudFormation](#)
- Pour plus d'informations sur l'ajout de politiques aux ensembles d'autorisations, voir [Création d'un ensemble d'autorisations](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment limiter les utilisateurs à un ensemble limité d' CodeDeploy actions et de ressources, voir [AWS politiques gérées \(prédéfinies\) pour CodeDeploy](#).

Le tableau suivant indique les actions qu'un CloudFormation modèle peut effectuer en votre nom et inclut des liens vers des informations supplémentaires sur les types de AWS ressources et leurs types de propriétés que vous pouvez ajouter à un CloudFormation modèle.

| Action                                                                                 | CloudFormation référence                         | Type de référence        |
|----------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------|
| Créez une CodeDeploy application.                                                      | <a href="#">AWS::CodeDeployer : :application</a> | CloudFormation ressource |
| Créez et spécifiez les détails d'un groupe de déploiement à utiliser pour déployer les | <a href="#">AWS::CodeDeploy::DeploymentGroup</a> | CloudFormation ressource |

| Action                                                                                                                                                                                                                                                                    | CloudFormation référence                          | Type de référence                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| révisions de votre application.<br><sup>1</sup>                                                                                                                                                                                                                           |                                                   |                                                                                                                                                                                                                                                                           |
| Créez un ensemble de règles de déploiement, de conditions de réussite du déploiement et de conditions d'échec du déploiement qui CodeDeploy seront utilisées lors d'un déploiement.                                                                                       | <a href="#">AWS::CodeDeploy::DeploymentConfig</a> | CloudFormation ressource                                                                                                                                                                                                                                                  |
| Créez une instance Amazon EC2.<br><sup>2</sup>                                                                                                                                                                                                                            | <a href="#">AWS::EC2::Instance</a>                | CloudFormation ressource                                                                                                                                                                                                                                                  |
| Utilisez le CloudFormation <code>AWS::CodeDeployBlueGreenTransform</code> et <code>AWS::CodeDeploy::BlueGreenHook</code> pour gérer les mises à jour de la pile, créer des ressources et transférer le trafic vers des CodeDeploy déploiements bleu/vert.<br><sup>3</sup> | <a href="#">AWS::CodeDeployBlueGreen</a>          | La transformation <code>AWS::CodeDeployBlueGreen</code> est une macro hébergée par CloudFormation                                                                                                                                                                         |
|                                                                                                                                                                                                                                                                           | <a href="#">AWS::CodeDeploy::BlueGreen</a>        | Le <code>AWS::CodeDeploy::BlueGreen</code> crochet est structuré comme une Hook ressource dans CloudFormation. Le hook inclut des paramètres qui remplacent votre CodeDeploy AppSpec fichier en pointant vers des hooks d'événements CodeDeploy du cycle de vie désignés. |

| Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | CloudFormation référence | Type de référence |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|-------------------|
| <p><sup>1</sup> Si vous spécifiez la version de la révision de l'application que vous souhaitez déployer dans le cadre du groupe de déploiement, votre révision cible sera déployée dès que le processus de provisionnement sera terminé. Pour plus d'informations sur la configuration des modèles, consultez les <a href="#">révisions de CodeDeploy DeploymentGroup déploiement S3Location</a> et de <a href="#">CodeDeploy DeploymentGroup déploiement GitHubLocation</a> dans le <a href="#">Guide</a> de l'AWS CloudFormation utilisateur.</p>                                                                                           |                          |                   |
| <p><sup>2</sup> Nous fournissons des modèles que vous pouvez utiliser pour créer des instances Amazon EC2 dans les régions prises en charge CodeDeploy . Pour plus d'informations sur l'utilisation de ces modèles, consultez la rubrique <a href="#">Création d'une instance Amazon EC2 pour CodeDeploy (modèle)CloudFormation</a>.</p>                                                                                                                                                                                                                                                                                                       |                          |                   |
| <p><sup>3</sup> Seuls les blue/green déploiements Amazon ECS sont pris en charge par cette configuration de déploiement. Pour plus d'informations sur les configurations de déploiement pour les blue/green déploiements Amazon ECS via CloudFormation, consultez <a href="#">Configurations de déploiement pour les déploiements CloudFormation bleu/vert (Amazon ECS)</a>. Pour plus d'informations sur les blue/green déploiements Amazon ECS via Amazon ECS CloudFormation et sur la manière de visualiser votre déploiement dans CodeDeploy, consultez <a href="#">Créez un blue/green déploiement Amazon ECS via CloudFormation</a>.</p> |                          |                   |

## Utilisation CodeDeploy avec Amazon Virtual Private Cloud

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos AWS ressources, vous pouvez établir une connexion privée entre votre VPC et CodeDeploy. Vous pouvez utiliser cette connexion pour que CodeDeploy communique avec vos ressources sur votre VPC sans passer par l'Internet public.

Amazon VPC est un AWS service que vous pouvez utiliser pour lancer des ressources AWS dans un réseau virtuel que vous définissez. Avec un VPC, vous contrôlez des paramètres réseau, tels que

la plage d'adresses IP, les sous-réseaux, les tables de routage et les passerelles réseau. Avec les points de terminaison VPC, le routage entre le VPC et les AWS services est géré par le AWS réseau, et vous pouvez utiliser les politiques IAM pour contrôler l'accès aux ressources des services.

Pour connecter votre VPC à CodeDeploy, vous définissez un point de terminaison VPC d'interface pour. CodeDeploy Un point de terminaison d'interface est une interface elastic network dotée d'une adresse IP privée qui sert de point d'entrée pour le trafic destiné à un AWS service pris en charge. Le point de terminaison fournit une connectivité fiable et évolutive CodeDeploy sans nécessiter de passerelle Internet, d'instance de traduction d'adresses réseau (NAT) ou de connexion VPN. Pour de plus amples informations, veuillez consulter [Qu'est-ce qu'Amazon VPC ?](#) dans le Guide de l'utilisateur Amazon VPC.

Les points de terminaison VPC d'interface sont alimentés par AWS PrivateLink une AWS technologie qui permet une communication privée entre les AWS services à l'aide d'une interface Elastic Network avec des adresses IP privées. Pour de plus amples informations, veuillez consulter [AWS PrivateLink](#).

Les étapes suivantes s'adressent aux utilisateurs d'Amazon VPC. Pour plus d'informations, consultez [Démarez](#) dans le Amazon VPC Guide de l'utilisateur.

## Disponibilité

CodeDeploy possède deux points de terminaison VPC : un pour les opérations d' CodeDeploy agent et un pour CodeDeploy les opérations d'API. Le tableau ci-dessous indique les AWS régions prises en charge pour chaque point de terminaison.

Pour plus d'informations et pour les points de terminaison FIPS, consultez la section Points de [AWS CodeDeploy terminaison](#) et quotas.

| Nom de la région           | Code région | Point final de l'agent | Point de terminaison d'API | Région conforme à la norme FIPS ? |
|----------------------------|-------------|------------------------|----------------------------|-----------------------------------|
| USA Est (Virginie du Nord) | us-east-1   | Oui                    | Oui                        | Oui                               |
| USA Est (Ohio)             | us-east-2   | Oui                    | Oui                        | Oui                               |

| Nom de la région               | Code région    | Point final de l'agent | Point de terminaison d'API | Région conforme à la norme FIPS ? |
|--------------------------------|----------------|------------------------|----------------------------|-----------------------------------|
| USA Ouest (Californie du Nord) | us-west-1      | Oui                    | Oui                        | Oui                               |
| USA Ouest (Oregon)             | us-west-2      | Oui                    | Oui                        | Oui                               |
| Afrique (Le Cap)               | af-south-1     | Oui                    | Non                        |                                   |
| Asie-Pacifique (Hong Kong)     | ap-east-1      | Oui                    | Oui                        |                                   |
| Asie-Pacifique (Hyderabad)     | ap-south-2     | Oui                    | Non                        |                                   |
| Asie-Pacifique (Jakarta)       | ap-southeast-3 | Oui                    | Non                        |                                   |
| Asie-Pacifique (Melbourne)     | ap-southeast-4 | Oui                    | Non                        |                                   |
| Asie-Pacifique (Mumbai)        | ap-south-1     | Oui                    | Oui                        |                                   |
| Asie-Pacifique (Osaka)         | ap-northeast-3 | Oui                    | Non                        |                                   |
| Asie-Pacifique (Séoul)         | ap-northeast-2 | Oui                    | Oui                        |                                   |
| Asie-Pacifique (Singapour)     | ap-southeast-1 | Oui                    | Oui                        |                                   |
| Asie-Pacifique (Sydney)        | ap-southeast-2 | Oui                    | Oui                        |                                   |

| Nom de la région       | Code région    | Point final de l'agent | Point de terminaison d'API | Région conforme à la norme FIPS ? |
|------------------------|----------------|------------------------|----------------------------|-----------------------------------|
| Asie-Pacifique (Tokyo) | ap-northeast-1 | Oui                    | Oui                        |                                   |
| Canada (Centre)        | ca-central-1   | Oui                    | Oui                        |                                   |
| Chine (Pékin)          | cn-north-1     | Oui                    | Non                        |                                   |
| Chine (Ningxia)        | cn-northwest-1 | Non                    | Non                        |                                   |
| Europe (Francfort)     | eu-central-1   | Oui                    | Oui                        |                                   |
| Europe (Irlande)       | eu-west-1      | Oui                    | Oui                        |                                   |
| Europe (Londres)       | eu-west-2      | Oui                    | Oui                        |                                   |
| Europe (Milan)         | eu-south-1     | Oui                    | Non                        |                                   |
| Europe (Paris)         | eu-west-3      | Oui                    | Oui                        |                                   |
| Europe (Espagne)       | eu-south-2     | Oui                    | Non                        |                                   |
| Europe (Stockholm)     | eu-north-1     | Oui                    | Oui                        |                                   |
| Europe (Zurich)        | eu-central-2   | Oui                    | Non                        |                                   |
| Israël (Tel Aviv)      | il-central-1   | Oui                    | Oui                        |                                   |
| Moyen-Orient (Bahreïn) | me-south-1     | Oui                    | Oui                        |                                   |
| Moyen-Orient (EAU)     | me-central-1   | Oui                    | Non                        |                                   |



| Nom de la région            | Code région   | Point final de l'agent | Point de terminaison d'API | Région conforme à la norme FIPS ? |
|-----------------------------|---------------|------------------------|----------------------------|-----------------------------------|
| Amérique du Sud (São Paulo) | sa-east-1     | Oui                    | Oui                        |                                   |
| AWS GovCloud (USA Est)      | us-gov-east-1 | Non                    | Non                        | Oui                               |
| AWS GovCloud (US-Ouest)     | us-gov-west-1 | Non                    | Non                        | Oui                               |

## Créez des points de terminaison VPC pour CodeDeploy

Pour commencer à utiliser CodeDeploy avec votre VPC, créez un point de terminaison VPC d'interface pour. CodeDeploy CodeDeploynécessite des points de terminaison distincts pour les opérations Git de l'agent et pour les opérations CodeDeploy d'API. Selon les besoins de votre entreprise, il se peut que vous ayez besoin de créer plusieurs points de terminaison de VPC. Lorsque vous créez un point de terminaison VPC pour CodeDeploy, choisissez AWS Services, puis dans Nom du service, choisissez l'une des options suivantes :

- `com.amazonaws. region.codedeploy` : Choisissez cette option si vous souhaitez créer un point de terminaison VPC pour les opérations d'API. CodeDeploy Par exemple, choisissez cette option si vos utilisateurs utilisent l' AWS CLI CodeDeploy API ou l'API AWS SDKs pour interagir CodeDeploy pour des opérations telles que `CreateApplicationGetDeployment`, et `ListDeploymentGroups`.
- `com.amazonaws. region.codedeploy-fips` : choisissez cette option si vous souhaitez créer un point de terminaison VPC pour les opérations d'API pour CodeDeploy le point de terminaison FIPS.
- `com.amazonaws. region. codedeploy-commands-secure`: Choisissez cette option si vous souhaitez créer un point de terminaison VPC pour les opérations des CodeDeploy agents. Vous devrez également le définir `true` dans le fichier `:enable_auth_policy:` de configuration de votre agent et y joindre les autorisations requises. Pour de plus amples informations, veuillez consulter [Configuration de l' CodeDeploy agent et des autorisations IAM](#).

Si vous utilisez des déploiements Lambda ou ECS, il vous suffit de créer un point de terminaison VPC pour `com.amazonaws.region.codedeploy`. Les clients utilisant des déploiements Amazon EC2 auront besoin de points de terminaison VPC pour `com.amazonaws.region.codedeploy` et `com.amazonaws.region.codedeploy-commands-secure`.

## Configuration de l' CodeDeploy agent et des autorisations IAM

Pour utiliser les points de terminaison Amazon VPC avec CodeDeploy, vous devez définir la valeur de `:enable_auth_policy:` to `true` dans le fichier de configuration de l'agent situé sur vos instances EC2 ou sur site. Pour plus d'informations sur le fichier de configuration de l'agent, consultez [CodeDeploy référence de configuration de l'agent](#).

Vous devez également ajouter les autorisations IAM suivantes à votre profil d'instance Amazon EC2 (si vous utilisez des instances Amazon EC2) ou à votre utilisateur ou rôle IAM (si vous utilisez des instances sur site).

```
{
 "Statement": [
 {
 "Action": [
 "codedeploy-commands-secure:GetDeploymentSpecification",
 "codedeploy-commands-secure:PollHostCommand",
 "codedeploy-commands-secure:PutHostCommandAcknowledgement",
 "codedeploy-commands-secure:PutHostCommandComplete"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Amazon VPC Guide de l'utilisateur.

## CodeDeploy référence du kit de ressources

La plupart des fichiers utilisés CodeDeploy sont stockés dans des compartiments Amazon S3 accessibles au public et AWS spécifiques à une région. Ces fichiers incluent les fichiers d'installation de l' CodeDeploy agent, les modèles et les exemples de fichiers d'application. Nous appelons cet ensemble de fichiers le kit de CodeDeploy ressources.

## Rubriques

- [Noms des compartiments du kit de ressources par région](#)
- [Contenu du kit de ressources](#)
- [Afficher la liste des fichiers du kit de ressources](#)
- [Téléchargez les fichiers du kit de ressources](#)

## Noms des compartiments du kit de ressources par région

Ce tableau répertorie les noms des *bucket-name* remplacements requis pour certaines procédures du guide. Il s'agit des noms des compartiments Amazon S3 qui contiennent les fichiers du kit de CodeDeploy ressources.

### Note

Pour accéder au compartiment Amazon S3 dans la région Asie-Pacifique (Hong Kong), vous devez activer la région dans votre AWS compte. Pour plus d'informations, consultez [la section Gestion des AWS régions](#).

| Nom de la région               | <i><b>Bucket-name</b></i> remplacement | Identifiant de région |
|--------------------------------|----------------------------------------|-----------------------|
| USA Est (Virginie du Nord)     | aws-codedeploy-us-east-1               | us-east-1             |
| USA Est (Ohio)                 | aws-codedeploy-us-east-2               | us-east-2             |
| USA Ouest (Californie du Nord) | aws-codedeploy-us-west-1               | us-west-1             |
| USA Ouest (Oregon)             | aws-codedeploy-us-west-2               | us-west-2             |
| Afrique (Le Cap)               | aws-codedeploy-af-south-1              | af-south-1            |
| Asie-Pacifique (Hong Kong)     | aws-codedeploy-ap-east-1               | ap-east-1             |
| Asie-Pacifique (Hyderabad)     | aws-codedeploy-ap-south-2              | ap-south-2            |
| Asie-Pacifique (Jakarta)       | aws-codedeploy-ap-southeast-3          | ap-southeast-3        |

| Nom de la région           | <i><b>Bucket-name</b></i> remplacement | Identifiant de région |
|----------------------------|----------------------------------------|-----------------------|
| Asie-Pacifique (Melbourne) | aws-codedeploy-ap-southeast-4          | ap-southeast-4        |
| Asie-Pacifique (Mumbai)    | aws-codedeploy-ap-south-1              | ap-south-1            |
| Asie-Pacifique (Osaka)     | aws-codedeploy-ap-northeast-3          | ap-northeast-3        |
| Asie-Pacifique (Séoul)     | aws-codedeploy-ap-northeast-2          | ap-northeast-2        |
| Asie-Pacifique (Singapour) | aws-codedeploy-ap-southeast-1          | ap-southeast-1        |
| Asie-Pacifique (Sydney)    | aws-codedeploy-ap-southeast-2          | ap-southeast-2        |
| Asie-Pacifique (Tokyo)     | aws-codedeploy-ap-northeast-1          | ap-northeast-1        |
| Canada (Centre)            | aws-codedeploy-ca-central-1            | ca-central-1          |
| Europe (Francfort)         | aws-codedeploy-eu-central-1            | eu-central-1          |
| Europe (Irlande)           | aws-codedeploy-eu-west-1               | eu-west-1             |
| Europe (Londres)           | aws-codedeploy-eu-west-2               | eu-west-2             |
| Europe (Milan)             | aws-codedeploy-eu-south-1              | eu-south-1            |
| Europe (Paris)             | aws-codedeploy-eu-west-3               | eu-west-3             |
| Europe (Espagne)           | aws-codedeploy-eu-south-2              | eu-south-2            |
| Europe (Stockholm)         | aws-codedeploy-eu-north-1              | eu-north-1            |
| Europe (Zurich)            | aws-codedeploy-eu-central-2            | eu-central-2          |
| Israël (Tel Aviv)          | aws-codedeploy-il-central-1            | il-central-1          |

| Nom de la région            | <b><i>Bucket-name</i></b> remplacement | Identifiant de région |
|-----------------------------|----------------------------------------|-----------------------|
| Middle East (Bahrain)       | aws-codedeploy-me-south-1              | me-south-1            |
| Moyen-Orient (EAU)          | aws-codedeploy-me-central-1            | me-central-1          |
| Amérique du Sud (São Paulo) | aws-codedeploy-sa-east-1               | sa-east-1             |
| AWS GovCloud (USA Est)      | aws-codedeploy-us-gov-est-1            | us-gov-east-1         |
| AWS GovCloud (US-Ouest)     | aws-codedeploy-us-gov-ouest-1          | us-gov-west-1         |

## Contenu du kit de ressources

Le tableau suivant répertorie les fichiers du kit de CodeDeploy ressources.

| Fichier                     | Description                                                                                                                                                                                                                                                     |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LATEST_VERSION              | Fichier utilisé par les mécanismes de mise à jour tels qu'Amazon EC2 Systems Manager pour déterminer la dernière version de l' CodeDeploy agent.                                                                                                                |
| VERSION                     | Le mécanisme de mise à jour automatique a été supprimé dans la version 1.1.0 de l' CodeDeploy agent et ce fichier n'est plus utilisé. Fichier que les agents CodeDeploy utilisent pour se mettre à jour lorsqu'ils sont en cours d'exécution sur des instances. |
| codedeploy-agent.noarch.rpm | L' CodeDeploy agent pour Amazon Linux et Red Hat Enterprise Linux (RHEL). Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que -1.0-0).                                                                    |

| Fichier                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>codedeploy-agent_all.deb</code>                 | L' CodeDeploy agent pour Ubuntu Server. Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que <code>_1.0-0</code> ).                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>codedeploy-agent.msi</code>                     | L' CodeDeploy agent pour Windows Server. Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que <code>-1.0-0</code> ).                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>install</code>                                  | Fichier que vous pouvez utiliser pour installer plus facilement l' CodeDeploy agent.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>CodeDeploy_SampleCF_Template.json</code>        | Un AWS CloudFormation modèle que vous pouvez utiliser pour lancer une à trois EC2 instances Amazon exécutant Amazon Linux ou Windows Server. Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que <code>-1.0.0</code> ).                                                                                                                                                                                                                                                                               |
| <code>CodeDeploy_SampleCF_ELB_Integration.json</code> | AWS CloudFormation Modèle que vous pouvez utiliser pour créer un exemple de site Web à charge équilibrée s'exécutant sur un serveur Web Apache. L'application est configurée pour couvrir toutes les zones de disponibilité de la région dans laquelle vous la créez. Ce modèle crée trois EC2 instances Amazon et un profil d'instance IAM pour accorder aux instances l'accès aux ressources d'Amazon S3, Amazon EC2 Auto Scaling et Elastic Load Balancing. CloudFormation Il crée également un équilibre ur de charge et un rôle CodeDeploy de service. |

| Fichier                       | Description                                                                                                                                                                                                                                                               |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SampleApp_ELB_Integration.zip | Exemple de révision d'application que vous pouvez déployer sur une EC2 instance Amazon enregistrée auprès d'un équilibreur de charge Elastic Load Balancing.                                                                                                              |
| SampleApp_Linux.zip           | Exemple de révision d'application que vous pouvez déployer sur une EC2 instance Amazon exécutant Amazon Linux ou sur une instance Ubuntu Server ou RHEL. Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que -1.0). |
| SampleApp_Windows.zip         | Exemple de révision d'application que vous pouvez déployer sur une instance Windows Server. Plusieurs fichiers peuvent avoir le même nom de fichier de base mais des versions différentes (telles que -1.0).                                                              |

## Afficher la liste des fichiers du kit de ressources

Pour afficher la liste des fichiers, utilisez la commande `aws s3 ls` pour votre région.

### Note

Les fichiers de chaque compartiment sont conçus pour fonctionner avec les ressources figurant dans la région correspondante.

- ```
aws s3 ls --recursive s3://aws-codedeploy-us-east-2 --region us-east-2
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-us-east-1 --region us-east-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-us-west-1 --region us-west-1
```

- `aws s3 ls --recursive s3://aws-codedeploy-us-west-2 --region us-west-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ca-central-1 --region ca-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-1 --region eu-west-1`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-2 --region eu-west-2`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-west-3 --region eu-west-3`
- `aws s3 ls --recursive s3://aws-codedeploy-eu-central-1 --region eu-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-il-central-1 --region il-central-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-east-1 --region ap-east-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-1 --region ap-northeast-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-2 --region ap-northeast-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-1 --region ap-southeast-1`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-2 --region ap-southeast-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-4 --region ap-southeast-4`
- `aws s3 ls --recursive s3://aws-codedeploy-ap-south-1 --region ap-south-1`
- `aws s3 ls --recursive s3://aws-codedeploy-sa-east-1 --region sa-east-1`

Téléchargez les fichiers du kit de ressources

Pour télécharger un fichier, utilisez la commande `aws s3 cp` pour votre région.

 Note

Veillez à utiliser le point (.) près de la fin. Cela permet de télécharger le fichier dans votre répertoire actif.

Par exemple, les commandes suivantes téléchargent un fichier individuel nommé `SampleApp_Linux.zip` à partir de l'un des dossiers `/samples/latest/` des compartiments :

- ```
aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2
```
- ```
aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1
```
- ```
aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1
```
- ```
aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2
```
- ```
aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3
```
- ```
aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1
```
- ```
aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1
```

- ```
aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4
```
- ```
aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1
```
- ```
aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1
```

Pour télécharger tous les fichiers, utilisez l'une des commandes suivantes pour votre région :

- ```
aws s3 cp --recursive s3://aws-codedeploy-us-east-2 . --region us-east-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-east-1 . --region us-east-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-west-1 . --region us-west-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-west-2 . --region us-west-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ca-central-1 . --region ca-central-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-eu-west-1 . --region eu-west-1
```

- `aws s3 cp --recursive s3://aws-codedeploy-eu-west-2 . --region eu-west-2`
- `aws s3 cp --recursive s3://aws-codedeploy-eu-west-3 . --region eu-west-3`
- `aws s3 cp --recursive s3://aws-codedeploy-eu-central-1 . --region eu-central-1`
- `aws s3 cp --recursive s3://aws-codedeploy-il-central-1 . --region il-central-1`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-east-1 . --region ap-east-1`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-1 . --region ap-northeast-1`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-2 . --region ap-northeast-2`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-1 . --region ap-southeast-1`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-2 . --region ap-southeast-2`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-4 . --region ap-southeast-4`
- `aws s3 cp --recursive s3://aws-codedeploy-ap-south-1 . --region ap-south-1`
- `aws s3 cp --recursive s3://aws-codedeploy-sa-east-1 . --region sa-east-1`

CodeDeploy quotas

Les tableaux suivants décrivent les quotas dans CodeDeploy.

Note

La limite d'heures d'exécution du déploiement sur place EC2 /On-Premises varie. Pour les configurations de déploiement personnalisées créées avant juin 2023, la limite est de 8 heures. Pour les configurations de déploiement personnalisées créées en juin 2023 ou ultérieurement, la limite est de 12 heures. Pour les configurations de déploiement prédéfinies, la limite est de 12 heures.

Name	Par défaut	Ajusté	Description
AWS Déploiement de Lambda effectué en quelques heures	Chaque région prise en charge : 50	Non	Nombre maximum d'heures pendant lesquelles un déploiement AWS Lambda peut être exécuté (48 heures pour la durée maximale entre le premier et le dernier changement de trafic, plus une heure pour chacun des deux hooks de cycle de vie possibles)
Applications associées par compte et par région	Chaque Région prise en charge : 1 000	Oui	Le nombre maximum de demandes associées à un AWS compte dans une seule région
Alarmes associées par groupe de déploiement	Chaque Région prise en charge : 50	Oui	Le nombre maximum d'alarmes associées à un groupe de déploiement
Groupes Auto Scaling dans un groupe de déploiement	Par région prise en charge : 10	Oui	Nombre maximum de groupes Amazon EC2 Auto Scaling dans un groupe de déploiement
Déploiements simultanés par compte	Chaque région prise en charge : 1 300	Oui	Nombre maximal de déploiements simultanés associés à un AWS compte. Chaque déploiement vers une EC2 instance Amazon agrandie d'un groupe Amazon EC2 Auto Scaling compte comme

Name	Par défaut	Ajuste	Description
			un seul déploiement simultané pour chaque application à laquelle l' EC2 instance est associée.
Déploiements simultanés par groupe de déploiement	Par région prise en charge : 1	Non	Nombre maximal de déploiements simultanés vers un groupe de déploiement. Cette limite empêche les déploiements simultanés de la même application vers le même groupe de déploiement.
Configurations de déploiement personnalisées par compte	Chaque région prise en charge : 50	Non	Nombre maximum de configurations de déploiement personnalisées associées à un AWS compte
Groupes de déploiement associés à une seule application	Chaque Région prise en charge : 1 000	Oui	Nombre maximal de groupes de déploiement associés à une application individuelle

Name	Par défaut	Ajusté	Description
EC2 blue/green /Déploiement sur site exécuté en quelques heures	Chaque région prise en charge : 109	Non	Nombre maximum d'heures pendant lesquelles un blue/green déploiement EC2 / On-Premises peut être exécuté (48 heures pour chacune des deux limites ci-dessus, plus une heure pour chacun des 13 événements du cycle de vie possibles)
EC2/Déploiement sur place sur site exécuté en quelques heures	Chaque région prise en charge : 12	Non	Nombre maximal d'heures pendant lesquelles un déploiement sur place EC2 /On-Premises peut être exécuté
Déclencheurs de notification d'événements dans un groupe de déploiement	Par région prise en charge : 10	Oui	Nombre maximal de déclencheurs de notification d'événements dans un groupe de déploiement
GitHub jetons de connexion par compte	Chaque région prise en charge : 25	Non	Nombre maximum de jetons de GitHub connexion pour un seul AWS compte
Heures entre la fin d'un déploiement et la fin des instances d'origine au cours d'un déploiement EC2 /On-Premises blue/green	Chaque région prise en charge : 48	Non	Nombre maximum d'heures entre la fin d'un déploiement et la fin des instances d'origine lors d'un déploiement EC2 / On-Premises blue/green

Name	Par défaut	Ajusté	Description
Heures entre le déploiement d'une révision et le transfert du trafic vers les instances de remplacement lors d'un déploiement EC2 /On-Premises blue/green	Chaque région prise en charge : 48	Non	Nombre maximum d'heures entre le déploiement d'une révision et le transfert du trafic vers les instances de remplacement lors d'un déploiement EC2 / On-Premises blue/green
Nombre d'instances par déploiement	us-east-1 : 2 000 Chacune des autres régions prises en charge : 1 000	Oui	Nombre maximal d'instances dans un déploiement unique
Quelques minutes qu'un blue/green déploiement peut attendre après un déploiement réussi avant de mettre fin aux instances du déploiement initial	Chaque région prise en charge : 2 800	Non	Nombre maximal de minutes qu'un blue/green déploiement peut attendre après un déploiement réussi avant de mettre fin aux instances du déploiement initial
Quelques minutes entre le premier et le dernier changement de trafic lors d'un déploiement AWS Lambda Canary ou linéaire	Chaque région prise en charge : 2 880	Non	Nombre maximal de minutes entre le premier et le dernier changement de trafic lors d'un déploiement AWS Lambda Canary ou linéaire

Name	Par défaut	Ajuste	Description
Quelques minutes avant l'échec d'un déploiement si aucun événement du cycle de vie ne démarre	Chaque région prise en charge : 5	Non	Nombre maximal de minutes avant l'échec d'un déploiement si aucun événement du cycle de vie ne démarre après (1) le déclenchement d'un déploiement à l'aide de la console ou de la commande create-deployment de la AWS CLI, ou (2) la fin de l'événement de cycle de vie précédent.
Nombre de groupes de déploiement pouvant être associés à un service Amazon ECS	Par région prise en charge : 1	Non	Nombre maximum de groupes de déploiement pouvant être associés à un service Amazon ECS
Nombre d'instances pouvant être transmises à l'action d' BatchGetOnPremisesInstances API	Chaque région prise en charge : 100	Non	Nombre maximal d'instances pouvant être transmises à l'action d' BatchGetOnPremisesInstances API
Nombre d'instances utilisées par des déploiements simultanés en cours par compte	us-east-1 : 3 000 Chacune des autres régions prises en charge : 1 000	Oui	Nombre maximal d'instances pouvant être utilisées par des déploiements simultanés en cours et associés à un compte

Name	Par défaut	Ajusté	Description
Nombre d'écouteurs pour un itinéraire de trafic lors d'un déploiement d'Amazon ECS	Par région prise en charge : 1	Non	Nombre maximal d'écouteurs pour un itinéraire de trafic lors d'un déploiement d'Amazon ECS
Quelques secondes avant qu'un événement du cycle de vie du déploiement échoue s'il n'est pas terminé	Chaque région prise en charge : 3 600 secondes	Non	Nombre de secondes maximal avant qu'un événement de cycle de vie de déploiement échoue s'il n'est pas terminé
Taille du nom du groupe de déploiement	Chaque région prise en charge : 100	Non	Nombre maximal de caractères dans un nom de groupe de déploiement
Taille de la clé de l'étiquette	Chaque région prise en charge : 128	Non	Nombre maximal de caractères dans une clé de balise
Taille de la valeur de l'étiquette	Chaque région prise en charge : 256	Non	Nombre maximal de caractères dans une valeur de balise
Balises dans un groupe de déploiement	Chaque Région prise en charge : 10	Non	Nombre maximal de balises dans un groupe de déploiement
Trafic pouvant être déplacé par incrément lors d'un déploiement AWS Lambda	Chaque région prise en charge : 99	Non	Pourcentage maximal de trafic pouvant être déplacé par incrément lors d'un déploiement AWS Lambda

Résolution des problèmes CodeDeploy

Utilisez les rubriques de cette section pour résoudre les problèmes et les erreurs que vous pourriez rencontrer lors de l'utilisation CodeDeploy.

Note

Vous pouvez identifier les causes de nombreux échecs de déploiement en passant en revue les fichiers journaux créés pendant le processus de déploiement. Pour des raisons de simplicité, nous vous recommandons d'utiliser Amazon CloudWatch Logs pour surveiller les fichiers journaux de manière centralisée au lieu de les consulter instance par instance. Pour plus d'informations, consultez [Monitoring Deployments with Amazon CloudWatch Tools](#).

Rubriques

- [Dépannage des problèmes généraux](#)
- [Résoudre les problèmes de déploiement d'EC2/sur site](#)
- [Résoudre les problèmes de déploiement d'Amazon ECS](#)
- [Résoudre les problèmes de déploiement AWS de Lambda](#)
- [Résolution des problèmes de groupe de déploiement](#)
- [Résolution des problèmes d'instance](#)
- [Résoudre les problèmes liés aux GitHub jetons](#)
- [Résoudre les problèmes liés à Amazon EC2 Auto Scaling](#)
- [Codes d'erreur pour AWS CodeDeploy](#)

Dépannage des problèmes généraux

Rubriques

- [Liste de contrôle de dépannage général](#)
- [CodeDeploy les ressources de déploiement ne sont prises en charge que dans certaines AWS régions](#)
- [Les procédures décrites dans ce guide ne correspondent pas à celles de la CodeDeploy console](#)
- [Les rôles IAM requis ne sont pas disponibles](#)

- [L'utilisation de certains éditeurs de texte pour créer des AppSpec fichiers et des scripts shell peut entraîner l'échec des déploiements.](#)
- [L'utilisation du Finder sur MacOS pour grouper une révision d'application peut entraîner l'échec des déploiements](#)

Liste de contrôle de dépannage général

Vous pouvez utiliser la liste de contrôle suivante pour dépanner un déploiement ayant échoué.

1. Consultez les rubriques [Afficher les détails CodeDeploy du déploiement](#) et [View Instance Details](#) pour déterminer pourquoi le déploiement a échoué. Si vous ne parvenez pas à déterminer la cause de l'échec, passez en revue les éléments de cette liste de contrôle.
2. Vérifiez que vous avez configuré les instances correctement :
 - L'instance a-t-elle été lancée avec une paire de clés EC2 spécifiée ? Pour plus d'informations, consultez les [paires de clés EC2](#) dans le guide de l'utilisateur Amazon EC2.
 - Le profil d'instance IAM approprié est-il attaché à l'instance ? Pour plus d'informations, consultez [Configurer une instance Amazon EC2 pour qu'elle fonctionne avec CodeDeploy](#) et [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#).
 - L'instance a-t-elle été balisée ? Pour plus d'informations, consultez la section [Utilisation des balises dans la console](#) dans le guide de l'utilisateur Amazon EC2.
 - L' CodeDeploy agent est-il installé, mis à jour et exécuté sur l'instance ? Pour de plus amples informations, veuillez consulter [Gestion des opérations des CodeDeploy agents](#). Pour vérifier quelle version de l'agent est installée, consultez [Déterminer la version de l' CodeDeploy agent](#).
3. Vérifiez les paramètres de l'application et du groupe de déploiement :
 - Pour vérifier les paramètres de votre application, consultez la rubrique [Consultez les détails de l'application avec CodeDeploy](#).
 - Pour vérifier les paramètres de votre groupe de déploiement, consultez la rubrique [Afficher les détails du groupe de déploiement avec CodeDeploy](#).
4. Confirmez que la révision d'application est configurée correctement :
 - Vérifiez le format de votre AppSpec fichier. Pour plus d'informations, consultez [Ajouter un fichier de spécification d'application à une révision pour CodeDeploy](#) et [CodeDeploy AppSpec référence de fichier](#).
 - Vérifiez votre compartiment ou GitHub référentiel Amazon S3 pour vérifier que la révision de votre application se trouve à l'emplacement prévu.

- Passez en revue les détails de la révision de votre CodeDeploy demande pour vous assurer qu'elle est correctement enregistrée. Pour plus d'informations, consultez [Consultez les détails de révision de l'application avec CodeDeploy](#).
 - Si vous effectuez un déploiement à partir d'Amazon S3, vérifiez dans votre compartiment Amazon S3 que vous avez CodeDeploy obtenu les autorisations nécessaires pour télécharger la révision de l'application. Pour plus d'informations sur les stratégies de compartiment, consultez la rubrique [Conditions préalables au déploiement](#).
 - Si vous effectuez le déploiement à partir de GitHub, vérifiez dans votre GitHub référentiel que CodeDeploy les autorisations nécessaires au téléchargement de la révision de l'application ont été accordées. Pour plus d'informations, consultez [Créez un déploiement avec CodeDeploy et GitHub authentication avec des applications dans CodeDeploy](#).
5. Vérifiez que le rôle de service est configuré correctement. Pour plus d'informations, consultez [Étape 2 : créer un rôle de service pour CodeDeploy](#).
6. Confirmez que vous avez suivi les étapes de la rubrique [Commencer avec CodeDeploy](#) pour :
- A octroyé à un utilisateur les autorisations appropriées.
 - installer ou mettre à niveau et configurer l'interface AWS CLI ;
 - Créez un profil d'instance IAM et un rôle de service.
- Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès pour AWS CodeDeploy](#).
7. Vérifiez que vous utilisez AWS CLI la version 1.6.1 ou ultérieure. Pour vérifier la version que vous avez installée, appelez `aws --version`.

Si vous ne parvenez toujours pas à dépanner votre déploiement ayant échoué, passez en revue les autres problèmes dans cette rubrique.

CodeDeploy les ressources de déploiement ne sont prises en charge que dans certaines AWS régions

Si vous ne voyez pas ou ne pouvez pas accéder aux applications, aux groupes de déploiement, aux instances ou à d'autres ressources de déploiement depuis la console AWS CLI ou depuis la CodeDeploy console, assurez-vous de faire référence à l'une des AWS régions répertoriées dans [Région et aux points de terminaison](#) dans. Références générales AWS

Les instances EC2 et les groupes Amazon EC2 Auto Scaling utilisés CodeDeploy dans les déploiements doivent être lancés et créés dans l'une de ces régions. AWS

Si vous utilisez le AWS CLI, exécutez la `aws configure` commande depuis le AWS CLI. Vous pouvez ensuite afficher et définir votre AWS région par défaut.

Si vous utilisez la CodeDeploy console, dans la barre de navigation, dans le sélecteur de région, choisissez l'une des AWS régions prises en charge.

Important

Pour utiliser les services de la région Chine (Pékin) ou de la région Chine (Ningxia), vous devez disposer d'un compte et d'informations d'identification pour ces régions. Les comptes et informations d'identification AWS des autres régions ne fonctionnent pas pour les régions de Pékin et du Ningxia, et vice versa.

Les informations relatives à certaines ressources pour les régions de Chine, telles que les noms des compartiments du kit de CodeDeploy ressources et les procédures d'installation des CodeDeploy agents, ne sont pas incluses dans cette édition du guide de CodeDeploy l'utilisateur.

Pour plus d'informations, consultez:

- [CodeDeploy](#) dans [Commencer AWS dans la région de Chine \(Pékin\)](#)
- CodeDeploy Guide de l'utilisateur pour les régions de Chine ([version anglaise](#) | [version chinoise](#))

Les procédures décrites dans ce guide ne correspondent pas à celles de la CodeDeploy console

Les procédures de ce guide sont rédigées pour la nouvelle conception de la console. Si vous utilisez l'ancienne version de la console, la plupart des concepts et des procédures de base de ce guide sont toujours applicables. Pour accéder à l'aide de la nouvelle console, cliquez sur l'icône d'information.

Les rôles IAM requis ne sont pas disponibles

Si vous vous fiez à un profil d'instance IAM ou à un rôle de service créé dans le cadre d'une AWS CloudFormation pile, si vous supprimez la pile, tous les rôles IAM sont également supprimés. C'est peut-être la raison pour laquelle le rôle IAM n'est plus affiché dans la console IAM et CodeDeploy ne

fonctionne plus comme prévu. Pour résoudre ce problème, vous devez recréer manuellement le rôle IAM supprimé.

L'utilisation de certains éditeurs de texte pour créer des AppSpec fichiers et des scripts shell peut entraîner l'échec des déploiements.

Certains éditeurs de texte introduisent des caractères non imprimables non conformes dans les fichiers. Si vous utilisez des éditeurs de texte pour créer ou modifier AppSpec des fichiers ou des fichiers de script shell destinés à être exécutés sur des instances Amazon Linux, Ubuntu Server ou RHEL, les déploiements basés sur ces fichiers risquent d'échouer. Lors de l'utilisation de CodeDeploy de ces fichiers lors d'un déploiement, la présence de ces caractères peut entraîner des échecs de validation de hard-to-troubleshoot AppSpec fichiers et d'exécution de scripts.

Dans la CodeDeploy console, sur la page des détails de l'événement pour le déploiement, choisissez Afficher les journaux. (Ou vous utilisez le AWS CLI pour appeler la [get-deployment-instance](#) commande.) Recherchez des erreurs telles que `invalid character`, `command not found`, ou `file not found`.

Pour résoudre ce problème, nous vous recommandons de procéder comme suit :

- N'utilisez pas d'éditeurs de texte qui introduisent des caractères non imprimables tels que les retours de transport (^M caractères) dans vos AppSpec fichiers et vos fichiers de script shell.
- Utilisez des éditeurs de texte qui affichent des caractères non imprimables tels que les retours de transport dans vos AppSpec fichiers et les fichiers de script shell, afin de détecter et de supprimer ceux qui pourraient être introduits. Pour obtenir des exemples de ce type d'éditeurs de texte, recherchez sur Internet les éditeurs de texte affichant les retours à la ligne.
- Utilisez des éditeurs de texte exécutés sur des instances Amazon Linux, Ubuntu Server ou RHEL pour créer des fichiers de script shell qui s'exécutent sur des instances Amazon Linux, Ubuntu Server ou RHEL. Pour obtenir des exemples de ce type d'éditeurs de texte, recherchez sur Internet les éditeurs de script shell Linux.
- Si vous devez utiliser un éditeur de texte sous Windows ou macOS pour créer des fichiers de script shell à exécuter sur des instances Amazon Linux, Ubuntu Server ou RHEL, utilisez un programme ou un utilitaire qui convertit le texte au format Windows ou macOS au format Unix. Pour obtenir des exemples de tels programmes et utilitaires, recherchez sur Internet « DOS vers UNIX » ou « Mac vers UNIX ». Veillez à tester les fichiers de script shell convertis sur les systèmes d'exploitation cibles.

L'utilisation du Finder sur MacOS pour grouper une révision d'application peut entraîner l'échec des déploiements

Les déploiements peuvent échouer si vous utilisez l'application d'interface utilisateur graphique (GUI) du Finder sur un Mac pour regrouper (compresser) un AppSpec fichier ainsi que les fichiers et scripts associés dans un fichier d'archive de révisions d'applications (.zip). Ceci tient au fait que Finder crée un dossier __MACOSX intermédiaire dans le fichier .zip et y place les fichiers composants. CodeDeploy ne peut pas trouver les fichiers composants, ce qui entraîne l'échec du déploiement.

Pour résoudre ce problème, nous vous recommandons d'utiliser la commande AWS CLI to call the [push](#), qui permet de compresser les fichiers des composants dans la structure attendue. Vous pouvez également utiliser le terminal au lieu de l'interface graphique utilisateur pour compresser les fichiers composants. Le terminal ne crée pas de dossier __MACOSX intermédiaire.

Résoudre les problèmes de déploiement d'EC2/sur site

Rubriques

- [CodeDeploy erreur d'identification CommandPoller manquante dans le plugin](#)
- [Le déploiement échoue avec le message « Échec de la validation du message PKCS7 signé »](#)
- [Le déploiement ou redéploiement des mêmes fichiers sur les mêmes emplacements d'instance échoue avec l'erreur « Le déploiement a échoué car un fichier spécifié existe déjà à l'emplacement »](#)
- [Les longs chemins de fichier provoquent des erreurs « Aucun fichier ou répertoire de ce type »](#)
- [Des processus de longue durée peuvent entraîner l'échec des déploiements](#)
- [Résolution des problèmes liés à un échec AllowTraffic du cycle de vie sans qu'aucune erreur ne soit signalée dans les journaux de déploiement](#)
- [Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic](#)
- [Résolution des problèmes liés à un échec du cycle de vie d'un DownloadBundle déploiement avec UnknownError : non ouvert pour lecture](#)
- [Résolution des erreurs d'événements de cycle de vie ignorés](#)
- [PowerShell Les scripts Windows n'utilisent pas la version 64 bits de Windows PowerShell par défaut](#)

 Note

Les causes de nombreux échecs de déploiement peuvent être identifiées en passant en revue les fichiers journaux créés pendant le processus de déploiement. Pour des raisons de simplicité, nous vous recommandons d'utiliser Amazon CloudWatch Logs pour surveiller les fichiers journaux de manière centralisée au lieu de les consulter instance par instance. Pour plus d'informations, voir [Afficher CodeDeploy les CloudWatch journaux dans la console Logs](#).

 Tip

Pour un runbook qui automatise de nombreuses tâches de dépannage liées aux déploiements EC2/sur site, consultez le manuel de référence du runbook [AWS Support-TroubleshootCodeDeploy](#) Systems Manager AWS Automation.

CodeDeploy erreur d'identification CommandPoller manquante dans le plugin

Si vous recevez une erreur similaire à

`InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile`, elle peut être provoquée par l'une des causes suivantes :

- Aucun profil d'instance IAM n'est associé à l'instance sur laquelle vous effectuez le déploiement.
- Les autorisations configurées pour votre profil d'instance IAM ne sont pas correctes.

Un profil d'instance IAM autorise l' CodeDeploy agent à communiquer avec CodeDeploy et à télécharger votre révision depuis Amazon S3. Pour les instances EC2, veuillez consulter [Gestion des identités et des accès pour AWS CodeDeploy](#). Pour les instances sur site, consultez [Working with On-Premises Instances](#).

Le déploiement échoue avec le message « Échec de la validation du message PKCS7 signé »

Ce message d'erreur indique que l'instance exécute une version de l' CodeDeploy agent qui prend uniquement en charge l'algorithme de hachage SHA-1. Support de l'algorithme de hachage SHA-2 a

été introduit dans la version 1.0.1.854 de l' CodeDeploy agent, publiée en novembre 2015. À compter du 17 octobre 2016, les déploiements échouent si une version de l' CodeDeploy agent antérieure à 1.0.1.854 est installée. Pour plus d'informations, voir pour [passer AWS à l'algorithme de SHA256 hachage pour les certificats SSL](#), [REMARQUE : retrait des agents CodeDeploy hôtes antérieurs à la version 1.0.1.85](#), et. [Mettre à jour l' CodeDeploy agent](#)

Le déploiement ou redéploiement des mêmes fichiers sur les mêmes emplacements d'instance échoue avec l'erreur « Le déploiement a échoué car un fichier spécifié existe déjà à l'emplacement »

Lorsque CodeDeploy vous essayez de déployer un fichier sur une instance mais qu'un fichier portant le même nom existe déjà à l'emplacement cible spécifié, le déploiement vers cette instance peut échouer. Vous pouvez recevoir le message d'erreur « Le déploiement a échoué car un fichier spécifié existe déjà à cet emplacement :*location-name*. » Cela est dû au fait que, pour chaque déploiement, CodeDeploy supprime d'abord tous les fichiers du déploiement précédent, lesquels sont répertoriés dans un fichier journal de nettoyage. Si certains fichiers des dossiers d'installation cibles ne sont pas répertoriés dans ce fichier de nettoyage, l' CodeDeploy agent interprète par défaut cela comme une erreur et échoue au déploiement.

Note

Sur les instances Amazon Linux, RHEL et Ubuntu Server, le fichier de nettoyage se trouve dans. /opt/codedeploy-agent/deployment-root/deployment-instructions/
Sur les instances Windows Server, l'emplacement est C:\ProgramData\Amazon
\CodeDeploy\deployment-instructions\.

Le moyen le plus simple d'éviter cette erreur est de spécifier une option autre que le comportement par défaut qui consiste à faire échouer le déploiement. Pour chaque déploiement, vous pouvez choisir de faire échouer le déploiement, de remplacer les fichiers non répertoriés dans le fichier de nettoyage ou de conserver les fichiers déjà présents sur l'instance.

L'option de remplacement est utile quand, par exemple, vous avez placé manuellement un fichier sur une instance après le dernier déploiement, mais que vous avez ensuite ajouté un fichier du même nom à la révision de l'application suivante.

Vous pouvez choisir l'option de conservation pour les fichiers que vous placez sur l'instance et dont vous souhaitez qu'ils fassent partie du prochain déploiement sans avoir à les ajouter au package

de révision de l'application. L'option de conservation est également utile si les fichiers de votre application se trouvent déjà dans votre environnement de production et que vous souhaitez les déployer CodeDeploy pour la première fois. Pour plus d'informations, consultez [Création d'un déploiement de plate-forme de calcul EC2/sur site \(console\)](#) et [Comportement d'annulation avec du contenu existant](#).

Résolution des erreurs de déploiement **The deployment failed because a specified file already exists at this location**

Si vous choisissez de ne pas spécifier la possibilité de remplacer ou de conserver le contenu qu'CodeDeploy détecte dans vos emplacements de déploiement cibles (ou si vous ne spécifiez pas d'option de déploiement pour traiter les contenus existants dans une commande de programmation), vous pouvez choisir de résoudre l'erreur.

Les informations suivantes s'appliquent uniquement si vous choisissez de ne pas conserver ou de remplacer le contenu.

Si vous essayez de redéploier des fichiers portant les mêmes noms et emplacements, le redéploiement a plus de chances de réussir si vous spécifiez le nom de l'application et le groupe de déploiement avec le même ID de groupe de déploiement sous-jacent que celui que vous avez utilisé auparavant. CodeDeploy utilise l'ID du groupe de déploiement sous-jacent pour identifier les fichiers à supprimer avant un redéploiement.

Le déploiement de nouveaux fichiers ou le redéploiement des mêmes fichiers aux mêmes emplacements sur les instances peut échouer pour les raisons suivantes :

- Vous avez spécifié un nom d'application différent pour un redéploiement de la même révision sur les mêmes instances. Le redéploiement échoue parce que, même si le nom du groupe de déploiement est le même, l'utilisation d'un nom d'application différent signifie qu'un autre ID de groupe de déploiement sous-jacent est utilisé.
- Vous avez supprimé et recréé un groupe de déploiement pour une application, puis essayé de redéploier la même révision dans le groupe de déploiement. Le redéploiement échoue car même si le nom du groupe de déploiement est identique, il CodeDeploy fait référence à un identifiant de groupe de déploiement sous-jacent différent.
- Vous avez supprimé une application et un groupe de déploiement dans CodeDeploy, puis vous avez créé un nouveau groupe d'applications et de déploiement portant les mêmes noms que ceux que vous avez supprimés. Après cela, vous avez essayé de redéploier une révision qui avait été déployée dans le groupe de déploiement précédent vers le nouveau groupe portant le même nom.

Le redéploiement échoue car même si les noms de l'application et du groupe de déploiement sont identiques, ils font CodeDeploy toujours référence à l'ID du groupe de déploiement que vous avez supprimé.

- Vous avez déployé une révision dans un groupe de déploiement, puis déployé la même révision dans un autre groupe de déploiement sur les mêmes instances. Le second déploiement échoue, car CodeDeploy référence un autre ID de groupe de déploiement sous-jacent.
- Vous avez déployé une révision dans un seul groupe de déploiement, puis déployé une autre révision dans un autre groupe de déploiement sur les mêmes instances. Il existe au moins un fichier portant le même nom et situé au même emplacement que le second groupe de déploiement tente de déployer. Le deuxième déploiement échoue car CodeDeploy le fichier existant n'est pas supprimé avant le début du deuxième déploiement. Les deux déploiements font référence à un groupe de déploiement différent. IDs
- Vous avez déployé une révision dans CodeDeploy, mais il existe au moins un fichier portant le même nom et se trouvant au même emplacement. Le déploiement échoue car, par défaut, CodeDeploy le fichier existant n'est pas supprimé avant le début du déploiement.

Pour traiter de telles situations, effectuez l'une des actions suivantes :

- Supprimez les fichiers des emplacements et des instances vers lesquels ils ont été précédemment déployés, puis réessayez d'effectuer le déploiement.
- Dans AppSpec le fichier de votre révision, que ce soit dans les événements du cycle de vie ApplicationStop ou de BeforeInstall déploiement, spécifiez un script personnalisé pour supprimer les fichiers dans tous les emplacements correspondant aux fichiers que votre révision est sur le point d'installer.
- Déployez ou redéployez les fichiers à des emplacements ou sur des instances qui ne faisaient pas partie des déploiements précédents.
- Avant de supprimer une application ou un groupe de déploiement, déployez une révision contenant un AppSpec fichier ne spécifiant aucun fichier à copier sur les instances. Pour le déploiement, spécifiez le nom de l'application et le nom du groupe de déploiement qui utilisent les mêmes applications et groupes de déploiement sous-jacents IDs que ceux que vous êtes sur le point de supprimer. (Vous pouvez utiliser la [get-deployment-group](#) commande pour récupérer l'ID du groupe de déploiement.) CodeDeploy utilise l'ID et le AppSpec fichier du groupe de déploiement sous-jacents pour supprimer tous les fichiers installés lors du précédent déploiement réussi.

Les longs chemins de fichier provoquent des erreurs « Aucun fichier ou répertoire de ce type »

Pour les déploiements sur des instances Windows, si le chemin du fichier `appspec.yml` contient un chemin de fichier supérieur à 260 caractères dans la section fichiers de votre fichier `appspec.yml`, les déploiements peuvent échouer avec une erreur similaire à la suivante :

```
No such file or directory @ dir_s_mkdir - C:\your-long-file-path
```

Cette erreur se produit parce que Windows n'autorise pas par défaut les chemins de fichiers supérieurs à 260 caractères, comme indiqué dans [la documentation de Microsoft](#).

Pour les versions 1.4.0 ou ultérieures de l' CodeDeploy agent, vous pouvez activer les longs chemins de fichiers de deux manières, en fonction du processus d'installation de l'agent :

Si l' CodeDeploy agent n'est pas encore installé :

1. Sur la machine sur laquelle vous comptez installer l' CodeDeploy agent, activez la clé de registre `LongPathsEnabled` Windows à l'aide de cette commande :

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"  
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Installez l' CodeDeploy agent. Pour de plus amples informations, veuillez consulter [Installation de l' CodeDeploy agent](#).

Si l' CodeDeploy agent a déjà été installé :

1. Sur la machine CodeDeploy agent, activez la clé de registre `LongPathsEnabled` Windows à l'aide de cette commande :

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"  
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Redémarrez l' CodeDeploy agent pour que la modification de la clé de registre soit prise en compte. Pour redémarrer l'agent, utilisez cette commande :

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Des processus de longue durée peuvent entraîner l'échec des déploiements

Pour les déploiements sur les instances Amazon Linux, Ubuntu Server et RHEL, si vous disposez d'un script de déploiement qui lance un processus de longue durée, vous CodeDeploy risquez de devoir attendre longtemps avant de voir le cycle de vie du déploiement échouer. La raison en est que si le processus s'exécute plus longtemps que les processus au premier plan et en arrière-plan sont censés durer dans cet événement, CodeDeploy arrête le déploiement et le considère comme ayant échoué, même si le processus s'exécute encore comme prévu.

Par exemple, une révision d'application contient deux fichiers à la racine, `after-install.sh` et `sleep.sh`. Son AppSpec fichier contient les instructions suivantes :

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

Le `after-install.sh` fichier s'exécute pendant l'événement du cycle de vie de l' `AfterInstall` application. Voici son contenu :

```
#!/bin/bash
/tmp/sleep.sh
```

Le fichier `sleep.sh` contient le code suivant, qui interrompt l'exécution du programme pendant trois minutes (180 secondes) et simule ainsi un processus de longue durée :

```
#!/bin/bash
sleep 180
```

Lorsque l'`after-install.sh` appelle `sleep.sh` `sleep.sh` démarre et s'exécute pendant trois minutes (180 secondes), soit deux minutes (120 secondes) au-delà de l'heure prévue CodeDeploy `sleep.sh` (et, par relation, `after-install.sh`) d'arrêt de l'exécution. Après un délai d'une minute (60 secondes), le déploiement CodeDeploy s'arrête et échoue lors de l'événement du cycle de vie

de l' `AfterInstall` application, même si `sleep.sh` son exécution se poursuit comme prévu. L'erreur suivante s'affiche :

```
Script at specified location: after-install.sh failed to complete in 60 seconds.
```

Vous ne pouvez pas ajouter simplement une esperluette (&) dans `after-install.sh` pour exécuter `sleep.sh` en arrière-plan.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

Cela peut laisser le déploiement en attente pendant la période d'expiration par défaut d'une heure du cycle de vie du déploiement, après quoi le déploiement CodeDeploy s'arrête et échoue lors de l'événement du cycle de vie de l' `AfterInstall` application, comme auparavant.

Dans `after-install.sh`, appelez `sleep.sh` comme suit, ce qui permet CodeDeploy de continuer après le début du processus :

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

Dans l'appel précédent, `sleep.sh` est le nom du processus que vous voulez commencer à exécuter en arrière-plan, en redirigeant stdout, stderr et stdin vers `/dev/null`.

Résolution des problèmes liés à un échec `AllowTraffic` du cycle de vie sans qu'aucune erreur ne soit signalée dans les journaux de déploiement

Dans certains cas, un blue/green déploiement échoue pendant l'événement du `AllowTraffic` cycle de vie, mais les journaux de déploiement n'indiquent pas la cause de l'échec.

Cet échec est généralement dû à des contrôles de santé mal configurés dans Elastic Load Balancing pour le Classic Load Balancer, l'Application Load Balancer ou le Network Load Balancer utilisés pour gérer le trafic du groupe de déploiement.

Pour résoudre le problème, passez en revue et corrigez les erreurs de configuration de la vérification de l'état pour l'équilibreur de charge.

Pour les équilibreurs de charge classiques, consultez la section [Configure Health Checks](#) dans le guide de l'utilisateur pour les équilibreurs de charge classiques et [ConfigureHealthCheck](#) dans la version 2012-06-01 de référence de l'API Elastic Load Balancing.

Pour les équilibreurs de charge d'application, consultez [la section Contrôles de santé de vos groupes cibles](#) dans le guide d'utilisation des équilibreurs de charge d'application.

Pour les Network Load Balancers, consultez [la section Health Checks for Your Target Groups](#) dans le guide de l'utilisateur du Network Load Balancer.

Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic

Au cours d'un déploiement, l' CodeDeploy agent exécute les scripts spécifiés pour ApplicationStop et AfterBlockTraffic dans le AppSpec fichier du précédent déploiement réussi. BeforeBlockTraffic (Tous les autres scripts sont exécutés à partir du AppSpec fichier dans le déploiement en cours.) Si l'un de ces scripts contient une erreur et ne s'exécute pas correctement, le déploiement peut échouer.

Les raisons possibles de ces échecs sont les suivantes :

- L' CodeDeploy agent trouve le *deployment-group-id_last_successful_install* fichier au bon emplacement, mais l'emplacement indiqué dans le *deployment-group-id_last_successful_install* fichier n'existe pas.

Sur les instances Amazon Linux, Ubuntu Server et RHEL, ce fichier doit exister dans /opt/codedeploy-agent/deployment-root/deployment-instructions.

Sur les instances Windows Server, ce fichier doit être stocké dans le C:\ProgramData\Amazon\CodeDeploy\deployment-instructions dossier.

- À l'emplacement indiqué dans le *deployment-group-id_last_successful_install* fichier, le AppSpec fichier n'est pas valide ou les scripts ne s'exécutent pas correctement.
- Le script contient une erreur qui ne peut pas être corrigée. Il ne réussit donc jamais à s'exécuter.

Utilisez la CodeDeploy console pour déterminer pourquoi un déploiement a pu échouer lors de l'un de ces événements. Sur la page des détails du déploiement, choisissez View Events. Sur la page de détails de l'instance, dans la AfterBlockTraffic ligne ApplicationStopBeforeBlockTraffic, ou, choisissez Afficher les journaux. Ou utilisez le AWS CLI pour appeler la [get-deployment-instance](#) commande.

Si l'échec est dû à un script du dernier déploiement réussi qui ne s'exécute jamais correctement, créez un déploiement et spécifiez que les ApplicationStop AfterBlockTraffic échecs doivent être ignorés. BeforeBlockTraffic Il existe deux façons de procéder :

- Utilisez la CodeDeploy console pour créer un déploiement. Sur la page Créer un déploiement, sous Défaillance d'un événement ApplicationStop du cycle de vie, sélectionnez Ne pas échouer le déploiement sur une instance si cet événement du cycle de vie de l'instance échoue.
- Utilisez le AWS CLI pour appeler la [create-deployment](#) commande et inclure l'--ignore-application-stop-failuresoption.

Lorsque vous déployez la révision de l'application à nouveau, le déploiement se poursuit même si l'un de ces trois événements du cycle de vie échoue. Si la nouvelle révision inclut des scripts fixes pour ces événements du cycle de vie, les déploiements futurs peuvent réussir sans appliquer ce correctif.

Résolution des problèmes liés à un échec du cycle de vie d'un DownloadBundle déploiement avec UnknownError : non ouvert pour lecture

Si vous essayez de déployer une révision d'application depuis Amazon S3 et que le déploiement échoue pendant l'événement du cycle de vie du DownloadBundle déploiement avec l'UnknownError: not opened for readingerreur suivante :

- Une erreur interne du service Amazon S3 s'est produite. Déployez de nouveau la révision d'application.
- Le profil d'instance IAM de votre instance EC2 n'est pas autorisé à accéder à la révision de l'application dans Amazon S3. Pour plus d'informations sur les politiques relatives aux compartiments Amazon S3, consultez [Transférer une révision CodeDeploy pour Amazon S3 \(déploiements EC2/sur site uniquement\)](#) et [Conditions préalables au déploiement](#).
- Les instances vers lesquelles vous déployez sont associées à une AWS région (par exemple, USA Ouest (Oregon)), mais le compartiment Amazon S3 qui contient la révision de l'application est associé à une autre AWS région (par exemple, USA Est (Virginie du Nord)). Assurez-vous que la révision de l'application se trouve dans un compartiment Amazon S3 associé à la même AWS région que les instances.

Sur la page de présentation des événements pour le déploiement, à la ligne Download bundle, choisissez View logs. Ou utilisez le AWS CLI pour appeler la [get-deployment-instance](#)commande.

Si cette erreur s'est produite, il devrait y avoir une erreur dans la sortie avec le code d'erreur `UnknownError` et le message d'erreur `not opened for reading`.

Pour déterminer la cause de cette erreur :

1. Activez la journalisation avec connexion sur au moins une des instances, puis déployez de nouveau la révision d'application.
2. Examinez le fichier de consignation avec connexion pour trouver l'erreur. Les messages d'erreur courants pour ce problème incluent l'expression « `access denied` ».
3. Une fois que vous avez examiné les fichiers journaux, nous vous recommandons de désactiver la journalisation avec connexion pour réduire la taille des fichiers journaux et la quantité d'informations sensibles qui peuvent apparaître dans la sortie en texte clair, sur l'instance, à l'avenir.

Pour plus d'informations sur la façon de trouver le fichier d'enregistrement des câbles et d'activer et de désactiver l'enregistrement des câbles, reportez-vous `:log_aws_wire:` à la section [Référence de configuration de CodeDeploy l'agent](#).

Résolution des erreurs d'événements de cycle de vie ignorés

Si tous les événements du cycle de vie d'un déploiement EC2 ou sur site sont ignorés, vous risquez de recevoir une erreur similaire à `The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group are experiencing problems. (Error code: HEALTH_CONSTRAINTS)` Voici certaines des causes et solutions possibles :

- L' `CodeDeploy` agent n'est peut-être pas installé ou ne s'exécute pas sur l'instance. Pour déterminer si l' `CodeDeploy` agent est en cours d'exécution :
 - Pour Amazon Linux, RHEL ou Ubuntu Server, exécutez la commande suivante :

```
systemctl status codedeploy-agent
```

- Pour Windows, exécutez la commande suivante :

```
powershell.exe -Command Get-Service -Name CodeDeployagent
```

Si l' CodeDeploy agent n'est pas installé ou n'est pas en cours d'exécution, consultez [Vérifiez que l' CodeDeploy agent est en cours d'exécution](#).

Il est possible que votre instance ne parvienne pas à atteindre le point de terminaison public CodeDeploy ou Amazon S3 via le port 443. Essayez l'une des actions suivantes :

- Attribuez une adresse IP publique à l'instance et utilisez sa table de routage pour autoriser l'accès Internet. Assurez-vous que le groupe de sécurité associé à l'instance autorise l'accès sortant sur le port 443 (HTTPS). Pour de plus amples informations, veuillez consulter [Protocole de communication et port pour l' CodeDeploy agent](#).
- Si une instance est provisionnée dans un sous-réseau privé, utilisez une passerelle NAT au lieu d'une passerelle Internet dans la table de routage. Pour plus d'informations, consultez [Passerelles NAT](#).
- Le rôle de service pour n'a CodeDeploy peut-être pas les autorisations requises. Pour configurer un rôle de service CodeDeploy, consultez [Étape 2 : créer un rôle de service pour CodeDeploy](#).
- Si vous utilisez un proxy HTTP, assurez-vous qu'il est spécifié dans le `:proxy_uri:` paramètre du fichier de configuration de l' CodeDeploy agent. Pour de plus amples informations, veuillez consulter [CodeDeploy référence de configuration de l'agent](#).
- La date et l'heure de signature de votre instance de déploiement pourraient ne pas correspondre à la date et l'heure de signature de votre demande de déploiement. Recherchez une erreur similaire à celle qui se trouve `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` dans le fichier journal de votre CodeDeploy agent. Si vous voyez cette erreur, suivez les étapes décrites dans [Résolution des erreurs de déploiement « InvalidSignatureException — Signature expirée : \[heure\] est maintenant antérieure à \[heure\] »](#). Pour de plus amples informations, veuillez consulter [Afficher les données du journal pour les déploiements CodeDeploy EC2/sur site](#).
- L' CodeDeploy agent peut s'arrêter parce qu'une instance manque de mémoire ou d'espace sur le disque dur. Essayez de réduire le nombre de déploiements archivés sur votre instance en mettant à jour le `max_revisions` paramètre dans la configuration de l' CodeDeploy agent. Si vous effectuez cette opération pour une instance EC2 et que le problème persiste, envisagez d'utiliser une instance plus grande. Par exemple, si votre type d'instance est `t2.small`, essayez d'utiliser une instance `t2.medium`. Pour plus d'informations, reportez-vous aux [Fichiers installés par l' CodeDeploy agent](#) sections [CodeDeploy référence de configuration de l'agent](#), et [Types d'instances](#).

- Il est possible qu'aucun profil d'instance IAM ne soit attaché à l'instance sur laquelle vous effectuez le déploiement ou qu'un profil d'instance IAM ne dispose pas des autorisations requises.
- Si aucun profil d'instance IAM n'est attaché à votre instance, créez-en un avec les autorisations requises, puis attachez-le.
- Si un profil d'instance IAM est déjà attaché à votre instance, assurez-vous qu'il dispose des autorisations requises.

Une fois que votre profil d'instance attaché est configuré avec les autorisations requises, redémarrez votre instance. Pour plus d'informations, consultez [Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2](#) la section [Rôles IAM pour Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2.

PowerShell Les scripts Windows n'utilisent pas la version 64 bits de Windows PowerShell par défaut

Si un PowerShell script Windows exécuté dans le cadre d'un déploiement repose sur des fonctionnalités 64 bits (par exemple, parce qu'il consomme plus de mémoire que ce qu'autorise une application 32 bits ou qu'il appelle des bibliothèques proposées uniquement dans une version 64 bits), le script risque de se bloquer ou de ne pas s'exécuter comme prévu. Cela est dû au fait que, par défaut, la version 32 bits de Windows est CodeDeploy utilisée PowerShell pour exécuter des PowerShell scripts Windows qui font partie d'une révision d'application.

Ajoutez le code suivant au début de tout script devant être exécuté avec la version 64 bits de Windows PowerShell :

```
# Are you running in 32-bit mode?
# (\SysWOW64\ = 32-bit mode)

if ($PSHOME -like "*SysWOW64*")
{
    Write-Warning "Restarting this script under 64-bit Windows PowerShell."

    # Restart this script under 64-bit Windows PowerShell.
    # (\SysNative\ redirects to \System32\ for 64-bit mode)

    & (Join-Path ($PSHOME -replace "SysWOW64", "SysNative") powershell.exe) -File `
        (Join-Path $PSScriptRoot $MyInvocation.MyCommand) @args

    # Exit 32-bit script.
```

```
Exit $LastExitCode
}

# Was restart successful?
Write-Warning "Hello from $PSHOME"
Write-Warning " (\SysWOW64\ = 32-bit mode, \System32\ = 64-bit mode)"
Write-Warning "Original arguments (if any): $args"

# Your 64-bit script code follows here...
# ...
```

Bien que les informations de chemin de fichier contenues dans ce code puissent sembler peu intuitives, Windows 32 bits PowerShell utilise un chemin tel que :

c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

Windows 64 bits PowerShell utilise un chemin tel que :

c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Résoudre les problèmes de déploiement d'Amazon ECS

Rubriques

- [Un délai d'attente survient pendant l'attente d'un ensemble de tâches de remplacement](#)
- [Un délai d'attente survient pendant l'attente de la poursuite d'une notification](#)
- [Le rôle IAM ne dispose pas de suffisamment d'autorisations](#)
- [Le délai de déploiement a expiré en attendant un rappel de statut](#)
- [Le déploiement a échoué car une ou plusieurs fonctions de validation des événements du cycle de vie ont échoué](#)
- [L'ELB n'a pas pu être mis à jour en raison de l'erreur suivante : Le groupe cible de l'ensemble de tâches principal doit être derrière l'écouteur](#)
- [Mon déploiement échoue parfois lorsque j'utilise Auto Scaling](#)
- [Seul ALB prend en charge le routage progressif du trafic, utilisez plutôt AllAtOnce le routage du trafic lorsque vous êtes un groupe create/update de déploiement](#)
- [Même si mon déploiement a réussi, l'ensemble de tâches de remplacement échoue aux tests de santé d'Elastic Load Balancing, et mon application est en panne](#)

- [Puis-je associer plusieurs équilibreurs de charge à un groupe de déploiement ?](#)
- [Puis-je effectuer des déploiements CodeDeploy bleu/vert sans équilibreur de charge ?](#)
- [Comment puis-je mettre à jour mon service Amazon ECS avec de nouvelles informations lors d'un déploiement ?](#)

Un délai d'attente survient pendant l'attente d'un ensemble de tâches de remplacement

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

```
The deployment timed out while waiting for the replacement task set to become healthy. This time out period is 60 minutes.
```

Cause possible : Cette erreur peut se produire en cas d'erreur dans votre fichier de définition de tâche ou dans d'autres fichiers liés au déploiement. Par exemple, s'il y a une faute de frappe dans le image champ de votre fichier de définition de tâche, Amazon ECS essaiera d'extraire la mauvaise image de conteneur et échouera continuellement, ce qui provoquera cette erreur.

Corrections possibles et prochaines étapes :

- Corrigez les erreurs typographiques et les problèmes de configuration dans votre fichier de définition des tâches et dans d'autres fichiers.
- Consultez l'événement de service Amazon ECS correspondant et découvrez pourquoi les tâches de remplacement ne se portent pas bien. Pour plus d'informations sur les événements Amazon ECS, consultez les [événements Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- Consultez la section relative au [dépannage d'Amazon ECS](#) du manuel Amazon Elastic Container Service Developer Guide pour détecter les erreurs liées aux messages de l'événement.

Un délai d'attente survient pendant l'attente de la poursuite d'une notification

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

```
The deployment timed out while waiting for a notification to continue. This time out period is n minutes.
```

Cause possible : Cette erreur peut se produire si vous avez spécifié un temps d'attente dans le champ Spécifiez quand rediriger le trafic lorsque vous avez créé votre groupe de déploiement, mais que le déploiement n'a pas pu se terminer avant l'expiration du temps d'attente.

Corrections possibles et prochaines étapes :

- Dans votre groupe de déploiement, définissez le moment où vous souhaitez rediriger le trafic sur une période plus longue et le redéployer. Pour de plus amples informations, veuillez consulter [Création d'un groupe de déploiement pour un déploiement Amazon ECS \(console\)](#).
- Dans votre groupe de déploiement, modifiez Spécifiez quand rediriger le trafic vers Réacheminer le trafic immédiatement et redéployer. Pour de plus amples informations, veuillez consulter [Création d'un groupe de déploiement pour un déploiement Amazon ECS \(console\)](#).
- Redéployez puis exécutez la [aws deploy continue-deployment](#) AWS CLI commande avec l'--deployment-wait-typeoption définie sur. READY_WAIT Assurez-vous d'exécuter cette commande avant l'expiration du délai indiqué dans Spécifier quand le trafic doit être redirigé.

Le rôle IAM ne dispose pas de suffisamment d'autorisations

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

The IAM role `role-arn` does not give you permission to perform operations in the following AWS service: AWSLambda.

Cause possible : Cette erreur peut se produire si vous avez spécifié une fonction Lambda dans la [Hookssection du AppSpec fichier](#), mais que vous n'avez pas CodeDeploy autorisé le service Lambda.

Solution possible : ajoutez l'lambda:InvokeFunctionautorisation au rôle CodeDeploy de service. Pour ajouter cette autorisation, ajoutez l'une des politiques AWS gérées suivantes au rôle : **AWSCodeDeployRoleForECS** ou **AWSCodeDeployRoleForECSLimited**. Pour plus d'informations sur ces politiques et sur la manière de les ajouter au rôle de CodeDeploy service, consultez [Étape 2 : créer un rôle de service pour CodeDeploy](#).

Le délai de déploiement a expiré en attendant un rappel de statut

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

The deployment timed out while waiting for a status callback. CodeDeploy expects a status callback within one hour after a deployment hook is invoked.

Cause possible : Cette erreur peut se produire si vous avez spécifié une fonction Lambda dans la [Hookssection du AppSpec fichier](#), mais que la fonction Lambda n'a pas pu appeler l'PutLifecycleEventHookExecutionStatusAPI nécessaire pour renvoyer un Succeeded statut ou à. Failed CodeDeploy

Corrections possibles et prochaines étapes :

- Ajoutez l'codedeploy:putlifecycleEventHookExecutionStatusautorisation au rôle d'exécution Lambda utilisé par la fonction Lambda que vous avez spécifiée dans le fichier. AppSpec Cette autorisation permet à la fonction Lambda de renvoyer un statut de Succeeded ou Failed à. CodeDeploy Pour plus d'informations sur le rôle d'exécution Lambda, consultez la section Rôle d'exécution [Lambda dans le Guide de l'utilisateur](#).AWS Lambda
- Vérifiez le code de votre fonction Lambda et les journaux d'exécution pour vous assurer que votre fonction Lambda appelle l'PutLifecycleEventHookExecutionStatusAPI afin de CodeDeploy savoir si CodeDeploy le test de validation du cycle de vie est ou. Succeeded Failed Pour plus d'informations sur l'putlifecycleEventHookExecutionStatusAPI, consultez [PutLifecycleEventHookExecutionStatus](#)la référence de l'AWS CodeDeploy API. Pour plus d'informations sur les journaux d'exécution Lambda, consultez la section Accès [CloudWatch aux journaux Amazon](#) pour. AWS Lambda

Le déploiement a échoué car une ou plusieurs fonctions de validation des événements du cycle de vie ont échoué

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

The deployment failed because one or more of the lifecycle event validation functions failed.

Cause possible : Cette erreur peut se produire si vous avez spécifié une fonction Lambda dans la [Hookssection du AppSpec fichier](#), mais que la fonction Lambda est revenue Failed à CodeDeploy son appel. PutLifecycleEventHookExecutionStatus Cet échec indique CodeDeploy que le test de validation du cycle de vie a échoué.

Étape suivante possible : consultez vos journaux d'exécution Lambda pour voir pourquoi le code du test de validation échoue. Pour plus d'informations sur les journaux d'exécution Lambda, consultez la section Accès [CloudWatch aux journaux Amazon](#) pour. AWS Lambda

L'ELB n'a pas pu être mis à jour en raison de l'erreur suivante : Le groupe cible de l'ensemble de tâches principal doit être derrière l'écouteur

Problème : le message d'erreur suivant s'affiche lors du déploiement de votre application Amazon ECS à l'aide de CodeDeploy :

```
The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
```

Cause possible : Cette erreur peut se produire si vous avez configuré un écouteur de test facultatif et qu'il est configuré avec le mauvais groupe cible. Pour plus d'informations sur l'écouteur de test intégré CodeDeploy, reportez-vous aux sections [Avant de commencer un déploiement Amazon ECS](#) et [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#). Pour plus d'informations sur les ensembles de tâches, consultez [TaskSet](#) le manuel Amazon Elastic Container Service API Reference et [describe-task-set](#) la section Amazon ECS du manuel AWS CLI Command Reference.

Solution possible : assurez-vous que l'écouteur de production et l'écouteur de test d'Elastic Load Balancing pointent tous deux vers le groupe cible qui gère actuellement vos charges de travail. Il y a trois endroits à vérifier :

- Dans Amazon EC2, dans les paramètres des écouteurs et des règles de votre équilibreur de charge. Pour plus d'informations, consultez la section [Écouteurs pour vos équilibreurs de charge d'application dans le guide de l'utilisateur pour les équilibreurs de charge d'application](#), ou [Écouteurs pour vos équilibreurs de charge réseau dans le guide de l'utilisateur pour les équilibreurs de charge réseau](#).
- Dans Amazon ECS, dans votre cluster, dans la configuration réseau de votre service. Pour plus d'informations, consultez les considérations relatives à [Application Load Balancer et Network Load Balancer](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- Dans CodeDeploy, dans les paramètres de votre groupe de déploiement. Pour de plus amples informations, veuillez consulter [Création d'un groupe de déploiement pour un déploiement Amazon ECS \(console\)](#).

Mon déploiement échoue parfois lorsque j'utilise Auto Scaling

Problème : vous utilisez Auto Scaling avec CodeDeploy et vous remarquez que vos déploiements échouent parfois. Pour plus d'informations sur les symptômes de ce problème, consultez la rubrique intitulée Pour les [services configurés pour utiliser le dimensionnement automatique des services et le type de blue/green déploiement, le dimensionnement automatique n'est pas bloqué pendant un déploiement, mais le déploiement peut échouer dans certaines circonstances](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Cause possible : ce problème peut se produire en cas CodeDeploy de conflit entre deux processus Auto Scaling.

Solution possible : Suspendez et reprenez les processus Auto Scaling pendant le CodeDeploy déploiement à l'aide de l'RegisterScalableTargetAPI (ou de la register-scalable-target AWS CLI commande correspondante). Pour plus d'informations, voir [Suspendre et reprendre le dimensionnement pour Application Auto Scaling](#) dans le Guide de l'utilisateur d'Application Auto Scaling.

Note

CodeDeploy Je ne peux pas appeler RegisterScaleableTarget directement. Pour utiliser cette API, vous devez configurer CodeDeploy pour envoyer une notification ou un événement à Amazon Simple Notification Service (ou Amazon CloudWatch). Vous devez ensuite configurer Amazon SNS (or CloudWatch) pour appeler une fonction Lambda, et configurer la fonction Lambda pour appeler l'API. RegisterScalableTarget L'RegisterScalableTargetAPI doit être appelée avec le SuspendedState paramètre défini sur true pour suspendre les opérations Auto Scaling et les false reprendre. La notification ou l'événement CodeDeploy envoyé doit se produire au démarrage d'un déploiement (pour déclencher les opérations de suspension d'Auto Scaling) ou lorsqu'un déploiement réussit, échoue ou s'arrête (pour déclencher les opérations de reprise d'Auto Scaling).

Pour plus d'informations sur la configuration CodeDeploy pour générer des notifications ou des CloudWatch événements Amazon SNS, consultez [Surveillance des déploiements avec Amazon CloudWatch Events](#). et. [Monitoring Deployments with Amazon SNS Event Notifications](#)

Seul ALB prend en charge le routage progressif du trafic, utilisez plutôt AllAtOnce le routage du trafic lorsque vous êtes un groupe create/update de déploiement

Problème : le message d'erreur suivant s'affiche lors de la création ou de la mise à jour d'un groupe de déploiement dans CodeDeploy :

Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group.

Cause possible : Cette erreur peut se produire si vous utilisez un Network Load Balancer et que vous essayez d'utiliser une configuration de déploiement prédéfinie autre que `CodeDeployDefault.ECSAllAtOnce`

Correctifs possibles :

- Modifiez votre configuration de déploiement prédéfinie en `CodeDeployDefault.ECSAllAtOnce`. Il s'agit de la seule configuration de déploiement prédéfinie prise en charge par les équilibreurs de charge réseau.

Pour plus d'informations sur les configurations de déploiement prédéfinies, consultez [Configurations de déploiement prédéfinies pour une plateforme de calcul Amazon ECS](#).

- Changez votre équilibreur de charge en un Application Load Balancer. Les équilibreurs de charge d'application prennent en charge toutes les configurations de déploiement prédéfinies. Pour plus d'informations sur la création d'un Application Load Balancer, consultez [Configuration d'un équilibreur de charge, de groupes cibles et d'écouteurs pour les déploiements CodeDeploy Amazon ECS](#)

Même si mon déploiement a réussi, l'ensemble de tâches de remplacement échoue aux tests de santé d'Elastic Load Balancing, et mon application est en panne

Problème : même si cela CodeDeploy indique que mon déploiement a réussi, l'ensemble de tâches de remplacement échoue aux tests de santé d'Elastic Load Balancing, et mon application est hors service.

Cause possible : ce problème peut se produire si vous avez effectué un CodeDeploy all-at-once déploiement et que votre ensemble de tâches de remplacement (vert) contient du code incorrect à l'origine de l'échec des tests de santé d'Elastic Load Balancing. Avec la configuration de all-at-once déploiement, les contrôles de santé de l'équilibreur de charge commencent à être exécutés sur l'ensemble de tâches de remplacement une fois que le trafic y a été transféré (c'est-à-dire après la survenue d'un événement CodeDeploy du AllowTraffic cycle de vie). C'est pourquoi les bilans de santé de l'ensemble de tâches de remplacement échoueront une fois que le trafic aura changé, mais pas avant. Pour plus d'informations sur les événements du cycle de vie CodeDeploy générés, consultez [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#).

Correctifs possibles :

- Changez la configuration de votre déploiement all-at-once en Canary ou Linear. Dans une configuration Canary ou linéaire, les contrôles de santé de l'équilibreur de charge commencent à être exécutés sur l'ensemble de tâches de remplacement pendant l'installation de votre application dans l'environnement de remplacement, et avant que le trafic ne soit déplacé (c'est-à-dire pendant l'événement Install du cycle de vie et avant l'AllowTraffic événement). En autorisant l'exécution des vérifications pendant l'installation de l'application, mais avant que le trafic ne soit transféré, un code d'application incorrect sera détecté et provoquera des échecs de déploiement avant que l'application ne soit rendue publique.

Pour plus d'informations sur la configuration des déploiements Canary ou Linear, consultez [Modifiez les paramètres du groupe de déploiement avec CodeDeploy](#).

Pour plus d'informations sur les événements CodeDeploy du cycle de vie qui s'exécutent lors d'un déploiement d'Amazon ECS, consultez [Que se passe-t-il lors d'un déploiement d'Amazon ECS](#).

 Note

Les configurations de déploiement Canary et linéaire ne sont prises en charge qu'avec les équilibreurs de charge d'application.

- Si vous souhaitez conserver votre configuration de all-at-once déploiement, configurez un écouteur de test et vérifiez l'état de santé de l'ensemble de tâches de remplacement à l'aide du hook du BeforeAllowTraffic cycle de vie. Pour de plus amples informations, veuillez consulter [Liste des hooks d'événements liés au cycle de vie pour un déploiement d'Amazon ECS](#).

Puis-je associer plusieurs équilibreurs de charge à un groupe de déploiement ?

Non Si vous souhaitez utiliser plusieurs équilibreurs de charge d'application ou de charge réseau, utilisez les mises à jour continues d'Amazon ECS au lieu de déploiements CodeDeploy bleu/vert. Pour plus d'informations sur les mises à jour continues, consultez la section [Mise à jour progressive](#) dans le manuel Amazon Elastic Container Service Developer Guide. Pour plus d'informations sur l'utilisation de plusieurs équilibreurs de charge avec Amazon ECS, consultez la section [Enregistrement de plusieurs groupes cibles auprès d'un service](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Puis-je effectuer des déploiements CodeDeploy bleu/vert sans équilibreur de charge ?

Non, vous ne pouvez pas effectuer de déploiements CodeDeploy bleu/vert sans équilibreur de charge. Si vous ne parvenez pas à utiliser un équilibreur de charge, utilisez plutôt la fonctionnalité de mises à jour continues d'Amazon ECS. Pour plus d'informations sur les mises à jour continues d'Amazon ECS, consultez la section [Mise à jour continue](#) du manuel Amazon Elastic Container Service Developer Guide.

Comment puis-je mettre à jour mon service Amazon ECS avec de nouvelles informations lors d'un déploiement ?

Pour CodeDeploy mettre à jour votre service Amazon ECS avec un nouveau paramètre pendant qu'il effectue un déploiement, spécifiez le paramètre dans la `resources` section du AppSpec fichier. Seuls quelques paramètres Amazon ECS sont pris en charge CodeDeploy, tels que le fichier de définition des tâches et les paramètres du nom du conteneur. Pour obtenir la liste complète des paramètres Amazon ECS CodeDeploy pouvant être mis à jour, consultez [AppSpec section « ressources » pour les déploiements Amazon ECS](#).

Note

Si vous devez mettre à jour votre service Amazon ECS avec un paramètre qui n'est pas pris en charge par CodeDeploy, effectuez les tâches suivantes :

1. Appelez l'`UpdateServiceAPI` d'Amazon ECS avec le paramètre que vous souhaitez mettre à jour. Pour obtenir la liste complète des paramètres pouvant être mis à jour, consultez [UpdateService](#) le manuel Amazon Elastic Container Service API Reference.

2. Pour appliquer la modification aux tâches, créez un nouveau blue/green déploiement Amazon ECS. Pour de plus amples informations, veuillez consulter [Création d'un déploiement d'Amazon ECS Compute Platform \(console\)](#).

Résoudre les problèmes de déploiement AWS de Lambda

Rubriques

- [AWS Lambda les déploiements échouent après l'arrêt manuel d'un déploiement Lambda pour lequel aucune annulation n'est configurée](#)

AWS Lambda les déploiements échouent après l'arrêt manuel d'un déploiement Lambda pour lequel aucune annulation n'est configurée

Dans certains cas, l'alias d'une fonction Lambda spécifié dans un déploiement peut faire référence à deux versions différentes de la fonction. Il en résulte que les tentatives suivantes de déploiement de la fonction Lambda échouent. Un déploiement Lambda peut passer dans cet état lorsqu'aucune restauration n'est configurée et qu'il est arrêté manuellement. Pour continuer, utilisez la AWS Lambda console pour vous assurer que la fonction n'est pas configurée pour transférer le trafic entre deux versions :

1. Connectez-vous à la AWS Lambda console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Dans le volet gauche, sélectionnez Functions.
3. Sélectionnez le nom de la fonction Lambda intégrée à votre CodeDeploy déploiement.
4. Dans Alias, choisissez l'alias utilisé dans votre CodeDeploy déploiement, puis sélectionnez Modifier.
5. Dans Alias pondéré, sélectionnez **none**. Cela garantit que l'alias n'est pas configuré pour déplacer un pourcentage ou un poids de trafic à plus d'une seule version. Notez la version sélectionnée dans Version.
6. Choisissez Enregistrer.
7. Ouvrez la CodeDeploy console et essayez de déployer la version affichée dans le menu déroulant de l'étape 5.

Résolution des problèmes de groupe de déploiement

Le balisage d'une instance dans le cadre d'un groupe de déploiement ne déploie pas automatiquement votre application sur la nouvelle instance

CodeDeploy ne déploie pas automatiquement votre application sur une instance nouvellement balisée. Vous devez créer un déploiement dans le groupe de déploiement.

Vous pouvez l'utiliser CodeDeploy pour activer les déploiements automatiques vers de nouvelles instances EC2 dans les groupes Amazon EC2 Auto Scaling. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy à Amazon EC2 Auto Scaling](#).

Résolution des problèmes d'instance

Rubriques

- [Les balises doivent être définies correctement](#)
- [AWS CodeDeploy l'agent doit être installé et exécuté sur les instances](#)
- [Les déploiements n'échouent pas pendant une heure lorsqu'une instance est mise hors service au cours d'un déploiement](#)
- [Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances](#)
- [Créez un nouveau fichier CodeDeploy journal s'il a été supprimé accidentellement](#)
- [Résolution des erreurs de déploiement « InvalidSignatureException — Signature expirée : \[heure\] est maintenant antérieure à \[heure\] »](#)

Les balises doivent être définies correctement

Utilisez la commande [list-deployment-instances](#) pour confirmer que les instances utilisées pour un déploiement sont balisées correctement. Si une instance EC2 est absente de la sortie, utilisez la console EC2 pour confirmer que les balises ont été définies sur l'instance. Pour plus d'informations, consultez la section [Utilisation des balises dans la console](#) dans le guide de l'utilisateur Amazon EC2.

Note

Si vous balisez une instance et que vous l'utilisez immédiatement CodeDeploy pour y déployer une application, l'instance risque de ne pas être incluse dans le déploiement. Cela

est dû au fait que plusieurs minutes peuvent être nécessaires avant de pouvoir lire les balises. Nous vous recommandons d'attendre au moins cinq minutes entre le moment où vous balisez une instance et celui où vous tentez d'y effectuer un déploiement.

AWS CodeDeploy l'agent doit être installé et exécuté sur les instances

Pour vérifier que l'agent CodeDeploy est installé et s'exécute sur une instance, consultez [Vérifiez que l'agent CodeDeploy est en cours d'exécution](#).

Pour installer, désinstaller ou réinstaller l'agent CodeDeploy, consultez [Installation de l'agent CodeDeploy](#).

Les déploiements n'échouent pas pendant une heure lorsqu'une instance est mise hors service au cours d'un déploiement

CodeDeploy fournit un délai d'une heure pour que chaque événement du cycle de vie du déploiement se déroule jusqu'à son terme. Cela fournit suffisamment de temps pour les scripts de longue durée.

Si les scripts ne s'exécutent pas complètement alors qu'un événement du cycle de vie est en cours (par exemple, si une instance est arrêtée ou si l'agent CodeDeploy est arrêté), l'affichage de l'état du déploiement comme Échec peut prendre jusqu'à une heure. Cela est vrai même si le délai d'expiration spécifié dans le script est inférieur à une heure. En effet, lorsque l'instance est arrêtée, l'agent CodeDeploy s'arrête et ne peut plus traiter d'autres scripts.

Si une instance est mise hors service entre des événements de cycle de vie ou avant le début de la première étape d'événement de cycle de vie, l'expiration du délai a lieu après seulement cinq minutes.

Analyse des fichiers journaux pour enquêter sur des échecs de déploiement sur les instances

Si le statut d'une instance dans le déploiement est différent de Succeeded, vous pouvez passer en revue les données du fichier journal de déploiement pour tenter d'identifier le problème. Pour de plus amples informations sur l'accès aux données du journal de déploiement, veuillez consulter [Afficher les données du journal pour les déploiements CodeDeploy EC2 sur site](#).

Créez un nouveau fichier CodeDeploy journal s'il a été supprimé accidentellement

Si vous supprimez accidentellement le fichier journal de déploiement sur une instance, CodeDeploy cela ne crée pas de fichier journal de remplacement. Pour créer un nouveau fichier journal, connectez-vous à l'instance, puis exécutez ces commandes :

Pour une instance Amazon Linux, Ubuntu Server ou RHEL, exécutez ces commandes dans cet ordre, une par une :

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Pour une instance Windows Server :

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Résolution des erreurs de déploiement « InvalidSignatureException — Signature expirée : [heure] est maintenant antérieure à [heure] »

CodeDeploy nécessite des références temporelles précises pour effectuer ses opérations. Si la date et l'heure de votre instance ne sont pas correctement définies, elles risquent de ne pas correspondre à la date de signature de votre demande de déploiement, qui est CodeDeploy rejetée.

Pour éviter des échecs de déploiement associés à des paramètres horaires incorrects, consultez les rubriques suivantes :

- [Réglage de l'heure pour votre instance Linux](#)
- [Réglage de l'heure pour une instance Windows](#)

Résoudre les problèmes liés aux GitHub jetons

GitHub OAuth Jeton non valide

CodeDeploy les applications créées après juin 2017 utilisent GitHub OAuth des jetons pour chaque AWS région. L'utilisation de jetons liés à des AWS régions spécifiques vous permet de mieux contrôler CodeDeploy les applications qui ont accès à un GitHub référentiel.

Si vous recevez une erreur de GitHub jeton, il se peut que vous ayez un ancien jeton qui n'est plus valide.

Pour corriger un GitHub OAuth jeton non valide

1. Supprimez l'ancien jeton en utilisant l'une des méthodes suivantes :
 - Pour supprimer l'ancien jeton à l'aide de l'API, utilisez [DeleteGitHubAccountToken](#).
 - Pour supprimer l'ancien jeton en utilisant AWS Command Line Interface :
 - a. Accédez à l'ordinateur sur lequel réside le jeton.
 - b. Assurez-vous que le AWS CLI est installé sur cet ordinateur. Pour les instructions d'installation, reportez-vous à la section [Installation, mise à jour et désinstallation du AWS CLI dans le](#) guide de l'AWS Command Line Interface utilisateur
 - c. Entrez la commande suivante sur l'ordinateur sur lequel réside le jeton :

aws delete-git-hub-account-token

Pour plus de détails sur la syntaxe des commandes, consultez [delete-git-hub-account-token](#).

2. Ajoutez un nouveau OAuth jeton. Pour de plus amples informations, veuillez consulter [Intégration CodeDeploy avec GitHub](#).

Nombre maximum de GitHub OAuth jetons dépassé

Lorsque vous créez un CodeDeploy déploiement, le nombre maximum de GitHub jetons autorisés est de 10. Si vous recevez un message d'erreur concernant les GitHub OAuth jetons, assurez-vous que vous disposez de 10 jetons ou moins. Si vous avez plus de 10 jetons, les premiers jetons qui ont été créés ne sont pas valides. Par exemple, si vous avez 11 jetons, le premier jeton que vous avez créé n'est pas valide. Si vous avez 12 jetons, les deux premiers jetons que vous avez créés ne sont

pas valides. Pour plus d'informations sur l'utilisation de l' API CodeDeploy pour supprimer d'anciens jetons, consultez [DeleteGitHubAccountToken](#).

Résoudre les problèmes liés à Amazon EC2 Auto Scaling

Rubriques

- [Résolution générale des problèmes liés à Amazon EC2 Auto Scaling](#)
- [« CodeDeployRole ne vous autorise pas à effectuer des opérations dans le AWS service suivant : AmazonAutoScaling » erreur](#)
- [Les instances d'un groupe Amazon EC2 Auto Scaling sont continuellement provisionnées et mises hors service avant qu'une révision ne puisse être déployée](#)
- [La mise hors service ou le redémarrage d'une instance Amazon EC2 Auto Scaling peut entraîner l'échec des déploiements](#)
- [Évitez d'associer plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling](#)
- [Les instances EC2 d'un groupe Amazon EC2 Auto Scaling ne se lancent pas et reçoivent l'erreur « Heartbeat Timeout »](#)
- [Des hooks de cycle de vie Amazon EC2 Auto Scaling mal adaptés peuvent entraîner l'arrêt ou l'échec des déploiements automatiques vers les groupes Amazon EC2 Auto Scaling](#)
- [Erreur « Le déploiement a échoué car aucune instance n'a été trouvée pour votre groupe de déploiement »](#)

Résolution générale des problèmes liés à Amazon EC2 Auto Scaling

Les déploiements vers des instances EC2 d'un groupe Amazon EC2 Auto Scaling peuvent échouer pour les raisons suivantes :

- Amazon EC2 Auto Scaling lance et met fin en permanence aux instances EC2. Si vous CodeDeploy ne parvenez pas à déployer automatiquement la révision de votre application, Amazon EC2 Auto Scaling lance et arrête en permanence les instances EC2.

Dissociez le groupe Amazon EC2 Auto Scaling du groupe de CodeDeploy déploiement ou modifiez la configuration de votre groupe Amazon EC2 Auto Scaling afin que le nombre d'instances souhaité corresponde au nombre actuel d'instances (empêchant ainsi Amazon EC2 Auto Scaling de lancer d'autres instances EC2). Pour plus d'informations, consultez notre section [Modifiez les paramètres du groupe de déploiement avec CodeDeploy](#) [Manual Scaling for Amazon EC2 Auto Scaling](#).

- L' CodeDeploy agent ne répond pas. L' CodeDeploy agent risque de ne pas être installé si les scripts d'initialisation (par exemple, les scripts cloud-init) qui s'exécutent immédiatement après le lancement ou le démarrage d'une instance EC2 mettent plus d'une heure à s'exécuter. CodeDeploy dispose d'un délai d'une heure pour que l' CodeDeploy agent réponde aux déploiements en attente. Pour résoudre ce problème, placez vos scripts d'initialisation dans votre révision d'application CodeDeploy.
- Une instance EC2 d'un groupe Amazon EC2 Auto Scaling redémarre lors d'un déploiement. Votre déploiement peut échouer si une instance EC2 est redémarrée pendant un déploiement ou si l' CodeDeploy agent est arrêté pendant le traitement d'une commande de déploiement. Pour de plus amples informations, veuillez consulter [La mise hors service ou le redémarrage d'une instance Amazon EC2 Auto Scaling peut entraîner l'échec des déploiements](#).
- Plusieurs révisions d'applications sont déployées simultanément sur la même instance EC2 dans un groupe Amazon EC2 Auto Scaling. Le déploiement simultané de plusieurs révisions d'application sur la même instance EC2 d'un groupe Amazon EC2 Auto Scaling peut échouer si l'un des déploiements comporte des scripts qui s'exécutent pendant plus de quelques minutes. Ne déployez pas plusieurs révisions d'applications sur les mêmes instances EC2 d'un groupe Amazon EC2 Auto Scaling.
- Un déploiement échoue pour les nouvelles instances EC2 lancées dans le cadre d'un groupe Amazon EC2 Auto Scaling. Dans ce scénario, l'exécution des scripts dans un déploiement peut empêcher le lancement d'instances EC2 dans le groupe Amazon EC2 Auto Scaling. (Les autres instances EC2 du groupe Amazon EC2 Auto Scaling peuvent sembler fonctionner normalement.) Pour résoudre ce problème, veillez à ce que tous les autres scripts se terminent en premier :
 - CodeDeploy l'agent n'est pas inclus dans votre AMI : si vous utilisez la `cfn-init` commande pour installer l' CodeDeploy agent lors du lancement d'une nouvelle instance, placez le script d'installation de l'agent à la fin de la `cfn-init` section de votre CloudFormation modèle.
 - CodeDeploy l'agent est inclus dans votre AMI : configurez l'AMI de manière à ce que l'agent soit dans son Stopped état lorsque l'instance est créée, puis incluez un script pour démarrer l'agent comme dernière étape dans votre bibliothèque de `cfn-init` scripts.

« CodeDeployRole ne vous autorise pas à effectuer des opérations dans le AWS service suivant : AmazonAutoScaling » erreur

Les déploiements qui utilisent un groupe Auto Scaling créé à l'aide d'un modèle de lancement nécessitent les autorisations suivantes. Elles s'ajoutent aux autorisations accordées par la politique `AWSCodeDeployRole` AWS gérée.

- `EC2:RunInstances`
- `EC2:CreateTags`
- `iam:PassRole`

Il se peut que vous receviez cette erreur si vous ne disposez pas ces autorisations. Pour plus d'informations [Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling](#), consultez [Création d'un modèle de lancement pour un groupe Auto Scaling](#) et [Permissions](#) dans le guide de l'utilisateur Amazon EC2 Auto Scaling.

Les instances d'un groupe Amazon EC2 Auto Scaling sont continuellement provisionnées et mises hors service avant qu'une révision ne puisse être déployée

Dans certains cas, une erreur peut empêcher un déploiement réussi sur des instances nouvellement provisionnées dans un groupe Amazon EC2 Auto Scaling. En conséquence, aucune instance n'est saine et aucun déploiement n'est réussi. Le déploiement ne pouvant pas être exécuté ou terminé, les instances sont terminées peu de temps après leur création. La configuration du groupe Amazon EC2 Auto Scaling entraîne ensuite le provisionnement d'un autre lot d'instances pour tenter de répondre aux exigences minimales en matière d'hôtes sains. Ce lot est également terminé et le cycle se poursuit.

Les causes possibles incluent :

- Échec des vérifications de l'état du groupe Amazon EC2 Auto Scaling.
- Erreur dans la révision d'application.

Pour résoudre ce problème, suivez les étapes suivantes :

1. Créez manuellement une instance EC2 qui ne fait pas partie du groupe Amazon EC2 Auto Scaling. Balisez l'instance avec une balise d'instance EC2 unique.
2. Ajoutez la nouvelle instance au groupe de déploiement affecté.
3. Déployez une nouvelle révision d'application sans erreur dans le groupe de déploiement.

Cela invite le groupe Amazon EC2 Auto Scaling à déployer la révision de l'application sur les futures instances du groupe Amazon EC2 Auto Scaling.

 Note

Après avoir confirmé que les déploiements sont réussis, supprimez l'instance que vous avez créée afin d'éviter des frais permanents sur votre AWS compte.

La mise hors service ou le redémarrage d'une instance Amazon EC2 Auto Scaling peut entraîner l'échec des déploiements

Si une instance EC2 est lancée via Amazon EC2 Auto Scaling, puis qu'elle est arrêtée ou redémarrée, les déploiements vers cette instance peuvent échouer pour les raisons suivantes :

- Au cours d'un déploiement en cours, un événement de scale-in ou tout autre événement de terminaison entraîne le détachement de l'instance du groupe Amazon EC2 Auto Scaling, puis son arrêt. Comme le déploiement ne peut pas être terminé, il échoue.
- L'instance est redémarrée, mais son démarrage prend plus de cinq minutes. CodeDeploy traite cela comme un délai d'attente. Le service considère tous les déploiements actuels et futurs sur l'instance comme ayant échoué.

Pour résoudre ce problème :

- En général, assurez-vous que tous les déploiements sont terminés avant la mise hors service ou le redémarrage de l'instance. Assurez-vous que tous les déploiements commencent après le démarrage ou le redémarrage de l'instance.
- Les déploiements peuvent échouer si vous spécifiez une Amazon Machine Image (AMI) basée sur Windows Server pour une configuration Amazon EC2 Auto Scaling et si vous utilisez le service EC2 Config pour définir le nom d'ordinateur de l'instance. Pour résoudre ce problème, dans l'AMI de base de Windows Server, sous l'onglet Général des propriétés du service EC2, désactivez l'option Définir le nom de l'ordinateur. Une fois que vous avez décoché cette case, ce comportement est désactivé pour toutes les nouvelles instances Amazon EC2 Auto Scaling de Windows Server lancées avec cette AMI de base Windows Server. Pour les instances Amazon EC2 Auto Scaling de Windows Server sur lesquelles ce comportement est activé, il n'est pas nécessaire de décocher cette case. Il vous suffit de redéployer les déploiements ayant échoué sur ces instances après qu'elles ont été redémarrées.

Évitez d'associer plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling

Il est recommandé d'associer un seul groupe de déploiement à chaque groupe Amazon EC2 Auto Scaling.

En effet, si Amazon EC2 Auto Scaling augmente le volume d'une instance comportant des hooks associés à plusieurs groupes de déploiement, il envoie des notifications pour tous les hooks en une seule fois. Cela entraîne le commencement simultané de plusieurs déploiements sur chaque instance. Lorsque plusieurs déploiements envoient des commandes à l'agent CodeDeploy en même temps, le délai de cinq minutes entre un événement du cycle de vie et le début du déploiement ou la fin de l'événement du cycle de vie précédent peut être atteint. Le cas échéant, le déploiement est mis en échec, même si un processus de déploiement se déroule par ailleurs sans difficulté.

Note

Le délai d'expiration par défaut d'un script dans un événement du cycle de vie est de 30 minutes. Vous pouvez remplacer le délai d'expiration par une autre valeur dans le AppSpec fichier. Pour de plus amples informations, veuillez consulter [Ajouter un AppSpec fichier pour un déploiement EC2/sur site](#).

Il n'est pas possible de contrôler l'ordre dans lequel les déploiements se produisent si plusieurs déploiements tentent de s'exécuter en même temps.

Enfin, si le déploiement sur une instance échoue, Amazon EC2 Auto Scaling met immédiatement fin à l'instance. Lorsque cette première instance s'arrête, les autres déploiements en cours d'exécution échouent progressivement. Étant donné que l'agent CodeDeploy dispose d'un délai d'une heure pour répondre aux déploiements en attente, le délai d'expiration de chaque instance peut prendre jusqu'à 60 minutes.

Pour plus d'informations sur Amazon EC2 Auto Scaling, [consultez Under the hood CodeDeploy : and Auto Scaling integration](#).

Les instances EC2 d'un groupe Amazon EC2 Auto Scaling ne se lancent pas et reçoivent l'erreur « Heartbeat Timeout »

Il se peut qu'un groupe Amazon EC2 Auto Scaling ne parvienne pas à lancer de nouvelles instances EC2, générant un message similaire au suivant :

Launching a new EC2 instance *<instance-Id>*. Status Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with token*<token-Id>* was abandoned: Heartbeat Timeout.

Ce message indique généralement l'une des situations suivantes :

- Le nombre maximum de déploiements simultanés associés à un AWS compte a été atteint. Pour plus d'informations sur les limites, consultez [CodeDeploy quotas](#).
- Le groupe Auto Scaling a essayé de lancer trop d'instances EC2 trop rapidement. Les appels d'API vers [RecordLifecycleActionHeartbeat](#) ou [CompleteLifecycleAction](#) pour chaque nouvelle instance ont été limités.
- Une application CodeDeploy a été supprimée avant que les groupes de déploiement associés ne soient mis à jour ou supprimés.

Lorsque vous supprimez une application ou un groupe de déploiement, vous CodeDeploy tentez de nettoyer tous les hooks Amazon EC2 Auto Scaling qui y sont associés, mais certains hooks peuvent subsister. Si vous exécutez une commande pour supprimer un groupe de déploiement, les hooks restants sont renvoyés dans la sortie. Toutefois, si vous exécutez une commande pour supprimer une application, les hooks restants n'apparaissent pas dans la sortie.

Par conséquent, la bonne pratique consiste à supprimer tous les groupes de déploiement associés à une application avant de supprimer l'application. Vous pouvez utiliser la sortie de la commande pour identifier les hooks du cycle de vie qui doivent être supprimés manuellement.

Si vous recevez un message d'erreur « Heartbeat Timeout », vous pouvez déterminer si les hooks de cycle de vie restants en sont la cause et résoudre le problème en procédant comme suit :

1. Effectuez l'une des actions suivantes :

- Appelez la [delete-deployment-group](#) commande pour supprimer le groupe de déploiement associé au groupe Auto Scaling à l'origine du délai d'expiration du rythme cardiaque.
- Appelez la [update-deployment-group](#) commande avec une liste vide non nulle de noms de groupes Auto Scaling pour détacher tous les hooks du cycle de vie Auto CodeDeploy Scaling gérés par Auto Scaling.

Par exemple, entrez la AWS CLI commande suivante :

```
aws deploy update-deployment-group --application-name my-example-app  
--current-deployment-group-name my-deployment-group --auto-scaling-  
groups
```

Autre exemple, si vous utilisez l' API CodeDeploy avec Java, appelez `UpdateDeploymentGroup` et définissez `autoScalingGroups` sur `new ArrayList<String>()`. Cela permet `autoScalingGroups` de définir une liste vide et de supprimer la liste existante. Ne l'utilisez pas `null`, ce qui est le cas par défaut, car cela reste tel `autoScalingGroups` quel, ce qui n'est pas ce que vous voulez.

Examinez la sortie de l'appel. Si la sortie contient une `hooksNotCleanedUp` structure avec une liste des hooks du cycle de vie d'Amazon EC2 Auto Scaling, il reste des hooks du cycle de vie.

2. Appelez la [describe-lifecycle-hooks](#) commande en spécifiant le nom du groupe Amazon EC2 Auto Scaling associé aux instances EC2 qui n'ont pas pu être lancées. Dans le résultat, recherchez l'un des éléments suivants :
 - Les noms des hooks du cycle de vie d'Amazon EC2 Auto Scaling correspondent à `hooksNotCleanedUp` la structure que vous avez identifiée à l'étape 1.
 - Noms des hooks du cycle de vie Amazon EC2 Auto Scaling contenant le nom du groupe de déploiement associé au groupe Auto Scaling défaillant.
 - Noms des hooks du cycle de vie d'Amazon EC2 Auto Scaling susceptibles d'avoir provoqué le délai d'expiration du déploiement. CodeDeploy
3. Si un hook appartient à l'une des catégories répertoriées à l'étape 2, appelez la [delete-lifecycle-hook](#) commande pour le supprimer. Spécifiez le groupe Amazon EC2 Auto Scaling et le hook du cycle de vie dans l'appel.

Important

Supprimez uniquement les hooks qui posent problème, comme indiqué à l'étape 2. Si vous supprimez des hooks viables, vos déploiements risquent d' échouer ou de ne pas être en mesure de déployer les révisions de votre application sur des instances EC2 redimensionnées.

4. Appelez la [create-deployment-group](#) commande [update-deployment-group](#) ou avec les noms de groupes Auto Scaling souhaités. CodeDeploy réinstalle les hooks Auto Scaling avec de nouveaux. UUIDs

 Note

Si vous dissociez un groupe Auto Scaling d'un groupe de CodeDeploy déploiement, les déploiements en cours vers le groupe Auto Scaling risquent d'échouer et les nouvelles instances EC2 redimensionnées par le groupe Auto Scaling ne recevront pas les révisions de votre application. CodeDeploy Pour que Auto Scaling fonctionne à nouveau CodeDeploy, vous devez rattacher le groupe Auto Scaling au groupe de déploiement et appeler un nouveau groupe `CreateDeployment` pour démarrer un déploiement à l'échelle du parc.

Des hooks de cycle de vie Amazon EC2 Auto Scaling mal adaptés peuvent entraîner l'arrêt ou l'échec des déploiements automatiques vers les groupes Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling CodeDeploy et utilisez des hooks de cycle de vie pour déterminer quelles révisions d'applications doivent être déployées sur quelles instances EC2 après leur lancement dans les groupes Amazon EC2 Auto Scaling. Les déploiements automatiques peuvent s'arrêter ou échouer si les hooks du cycle de vie et les informations les concernant ne correspondent pas exactement dans Amazon EC2 Auto Scaling et. CodeDeploy

Si les déploiements vers un groupe Amazon EC2 Auto Scaling échouent, vérifiez si les noms des crochets du cycle de vie dans Amazon EC2 Auto Scaling correspondent. CodeDeploy Si ce n'est pas le cas, utilisez ces appels de AWS CLI commande.

Tout d'abord, obtenez la liste des noms des hooks du cycle de vie pour le groupe Amazon EC2 Auto Scaling et le groupe de déploiement :

1. Appelez la [describe-lifecycle-hooks](#) commande en spécifiant le nom du groupe Amazon EC2 Auto Scaling associé au groupe de déploiement dans. CodeDeploy Dans la sortie, dans la liste `LifecycleHooks`, notez chaque valeur `LifecycleHookName`.
2. Appelez la [get-deployment-group](#) commande en spécifiant le nom du groupe de déploiement associé au groupe Amazon EC2 Auto Scaling. Dans la sortie, dans la `autoScalingGroups` liste, recherchez chaque élément dont le nom correspond au nom du groupe Amazon EC2 Auto Scaling, puis notez la valeur `hook` correspondante.

Comparez alors les deux ensembles de noms de hooks de cycle de vie. S'ils correspondent exactement, caractère par caractère, le problème est ailleurs. Vous pouvez essayer d'autres étapes de dépannage d'Amazon EC2 Auto Scaling décrites ailleurs dans cette section.

Toutefois, si les deux ensembles de noms de hooks de cycle de vie ne correspondent pas exactement, caractère par caractère, procédez comme suit :

1. Si des noms de hooks de cycle de vie figurent dans la sortie de la commande `describe-lifecycle-hooks` qui ne figurent pas également dans la sortie de la commande `get-deployment-group`, procédez comme suit :
 - a. Pour chaque nom de hook du cycle de vie indiqué dans la sortie de `describe-lifecycle-hooks` commande, appelez la [delete-lifecycle-hook](#) commande.
 - b. Appelez la [update-deployment-group](#) commande en spécifiant le nom du groupe Amazon EC2 Auto Scaling d'origine. CodeDeploy crée de nouveaux hooks de cycle de vie de remplacement dans le groupe Amazon EC2 Auto Scaling et associe les crochets de cycle de vie au groupe de déploiement. Les déploiements automatiques devraient maintenant reprendre à mesure que de nouvelles instances sont ajoutées au groupe Amazon EC2 Auto Scaling.
2. Si des noms de hooks de cycle de vie figurent dans la sortie de la commande `get-deployment-group`, mais ne figurent pas dans la sortie de la commande `describe-lifecycle-hooks`, procédez comme suit :
 - a. Appelez la [update-deployment-group](#) commande, mais ne spécifiez pas le nom du groupe Amazon EC2 Auto Scaling d'origine.
 - b. Appelez à nouveau la `update-deployment-group` commande, mais spécifiez cette fois le nom du groupe Amazon EC2 Auto Scaling d'origine. CodeDeploy recrée les hooks du cycle de vie manquants dans le groupe Amazon EC2 Auto Scaling. Les déploiements automatiques devraient maintenant reprendre à mesure que de nouvelles instances sont ajoutées au groupe Amazon EC2 Auto Scaling.

Une fois que les deux ensembles de noms de hooks du cycle de vie correspondent exactement, caractère par caractère, les révisions de l'application doivent être déployées à nouveau, mais uniquement sur les nouvelles instances au fur et à mesure qu'elles sont ajoutées au groupe Amazon EC2 Auto Scaling. Les déploiements ne sont pas effectués automatiquement sur des instances qui font déjà partie du groupe Amazon EC2 Auto Scaling.

Erreur « Le déploiement a échoué car aucune instance n'a été trouvée pour votre groupe de déploiement »

Lisez cette section si le message d' CodeDeploy erreur suivant s'affiche :

The deployment failed because no instances were found for your deployment group. Check your deployment group settings to make sure the tags for your EC2 instances or Auto Scaling groups correctly identify the instances you want to deploy to, and then try again.

Les causes possibles de cette erreur sont les suivantes :

1. Les paramètres de votre groupe de déploiement incluent des balises incorrectes pour les instances EC2, les instances sur site ou les groupes Auto Scaling. Pour résoudre ce problème, vérifiez que vos balises sont correctes, puis redéployez votre application.
2. Votre flotte s'est agrandie après le début du déploiement. Dans ce scénario, vous voyez des instances saines dans l'InService état de votre flotte, mais vous voyez également l'erreur ci-dessus. Pour résoudre ce problème, redéployez votre application.
3. Votre groupe Auto Scaling n'inclut aucune instance présente dans InService cet état. Dans ce scénario, lorsque vous essayez d'effectuer un déploiement à l'échelle du parc, le déploiement échoue avec le message d'erreur ci-dessus car il CodeDeploy faut qu'au moins une instance soit dans cet état. InService Il existe de nombreuses raisons pour lesquelles vous n'avez peut-être aucune instance dans l'InServiceÉtat. Quelques-uns d'entre eux incluent :
 - Vous avez planifié (ou configuré manuellement) la taille du groupe Auto Scaling comme suit 0 :
 - Auto Scaling a détecté des instances EC2 défectueuses (par exemple, les instances EC2 présentaient des défaillances matérielles), les a donc toutes annulées, n'en laissant aucune dans l'InServiceétat.
 - Au cours d'un événement de 0 scale-out de à1, a CodeDeploy déployé une révision précédemment réussie (appelée dernière révision réussie) qui était devenue défectueuse depuis le dernier déploiement. Cela a entraîné l'échec du déploiement sur l'instance redimensionnée, ce qui a entraîné l'annulation de l'instance par Auto Scaling, ne laissant aucune instance dans cet état. InService

Si vous constatez qu'aucune instance n'est présente dans InService cet état, résolvez le problème comme décrit dans la procédure suivante, [To troubleshoot the error if there are no instances in the InService state](#).

Pour résoudre l'erreur s'il n'y a aucune instance dans l'état InService

1. Dans la console Amazon EC2, vérifiez le paramètre Capacité souhaitée. S'il est égal à zéro, définissez-le sur un nombre positif. Attendez que l'instance existeInService, ce qui signifie que le déploiement a réussi. Vous avez résolu le problème et pouvez ignorer les étapes restantes de cette procédure de dépannage. Pour plus d'informations sur la définition du paramètre Desired Capacity, consultez la section [Setting capacity limits on your Auto Scaling group](#) dans le manuel Amazon EC2 Auto Scaling User Guide.
2. Si Auto Scaling continue d'essayer de lancer de nouvelles instances EC2 pour atteindre la capacité souhaitée mais ne parvient jamais à le faire évoluer, cela est généralement dû à un échec du hook du cycle de vie d'Auto Scaling. Résolvez ce problème comme suit :
 - a. Pour savoir quel événement Auto Scaling Lifecycle Hook échoue, consultez la section [Vérification d'une activité de dimensionnement pour un groupe Auto Scaling](#) dans le guide de l'utilisateur Amazon EC2 Auto Scaling.
 - b. Si le nom du hook défaillant est `CodeDeploy-managed-automatic-launch-deployment-hook-DEPLOYMENT_GROUP_NAME`, accédez au groupe de déploiement CodeDeploy, recherchez le groupe de déploiement et recherchez le déploiement ayant échoué lancé par Auto Scaling. Déterminez ensuite pourquoi le déploiement a échoué.
 - c. Si vous comprenez pourquoi le déploiement a échoué (par exemple, des CloudWatch alarmes se sont produites) et que vous pouvez résoudre le problème sans modifier la révision, faites-le maintenant.
 - d. Si, après enquête, vous déterminez que CodeDeploy la dernière révision réussie n'est plus saine et qu'il n'y a aucune instance saine dans votre groupe Auto Scaling, vous êtes dans un scénario de blocage du déploiement. Pour résoudre ce problème, vous devez corriger la mauvaise CodeDeploy révision en supprimant temporairement le hook CodeDeploy du cycle de vie du groupe Auto Scaling, puis en le réinstallant et en redéploiant une nouvelle (bonne) révision. Pour obtenir des instructions, consultez :
 - [To fix the deployment deadlock issue \(CLI\)](#)
 - [To fix the deployment deadlock issue \(console\)](#)

Pour résoudre le problème de blocage du déploiement (CLI)

1. (Facultatif) Bloquez les CI/CD pipelines à l'origine de l'erreur CodeDeploy afin d'éviter des déploiements inattendus pendant que vous résolvez ce problème.

2. Prenez note de votre DesiredCapacityparamètre Auto Scaling actuel :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name  
ASG_NAME
```

Vous devrez peut-être revenir à ce chiffre à la fin de cette procédure.

3. Définissez le DesiredCapacityparamètre Auto Scaling sur1. Ceci est facultatif si la capacité souhaitée était supérieure 1 à la capacité initiale. En le réduisant à1, l'instance mettra moins de temps à provisionner et à déployer ultérieurement, ce qui accélère le dépannage. Si la capacité souhaitée par Auto Scaling était initialement définie sur0, vous devez l'augmenter à1. Cela est obligatoire.

```
aws autoscaling set-desired-capacity -- auto-scaling-group-name ASG_NAME --desired-capacity 1
```

 Note

Les étapes restantes de cette procédure supposent que vous avez défini votre valeur DesiredCapacitysur1.

À ce stade, Auto Scaling tente d'effectuer une mise à l'échelle jusqu'à une seule instance. Ensuite, comme le hook CodeDeploy ajouté est toujours présent, CodeDeploy essaie de se déployer ; le déploiement échoue ; Auto Scaling annule l'instance ; et Auto Scaling essaie de relancer une instance pour atteindre la capacité souhaitée d'une instance, mais échoue à nouveau. Vous êtes dans une boucle d'annulation/relance.

4. Désenregistrez le groupe Auto Scaling du groupe de déploiement :

 Warning

La commande suivante lancera une nouvelle instance EC2 sans logiciel. Avant d'exécuter la commande, assurez-vous qu'une InService instance Auto Scaling n'exécutant aucun logiciel est acceptable. Par exemple, assurez-vous que l'équilibreur de charge associé à l'instance n'envoie pas de trafic vers cet hôte sans logiciel.

 Important

Utilisez la CodeDeploy commande ci-dessous pour retirer le crochet. Ne supprimez pas le crochet via le service Auto Scaling, car le retrait ne sera pas reconnu par CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME  
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-  
groups
```

Après avoir exécuté cette commande, voici ce qui se produit :

- a. CodeDeploy désenregistre le groupe Auto Scaling du groupe de déploiement.
 - b. CodeDeploy supprime le hook du cycle de vie Auto Scaling du groupe Auto Scaling.
 - c. Le hook à l'origine de l'échec du déploiement n'étant plus présent, Auto Scaling annule l'instance EC2 existante et en lance immédiatement une nouvelle pour l'adapter à la capacité souhaitée. La nouvelle instance devrait bientôt passer à InService l'état. La nouvelle instance n'inclut aucun logiciel.
5. Attendez que l'instance EC2 entre dans l'InService état. Pour vérifier son état, utilisez la commande suivante :

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names  
ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
```

6. Ajoutez le hook à l'instance EC2 :

 Important

Utilisez la CodeDeploy commande ci-dessous pour ajouter le crochet. N'utilisez pas le service Auto Scaling pour ajouter le hook, car l'ajout ne sera pas reconnu par CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME  
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-  
groups ASG_NAME
```

Après avoir exécuté cette commande, voici ce qui se produit :

- a. CodeDeploy réinstalle le hook du cycle de vie Auto Scaling sur l'instance EC2
 - b. CodeDeploy réenregistre le groupe Auto Scaling auprès du groupe de déploiement.
7. Créez un déploiement à l'échelle du parc à l'aide de l'Amazon S3 ou de GitHub la version dont vous savez qu'elle est saine et que vous souhaitez utiliser.

Par exemple, si la révision est un fichier .zip dans un compartiment Amazon S3 appelé my-revision-bucket avec une clé d'objet de httpd_app.zip, entrez la commande suivante :

```
aws deploy create-deployment --application-name APPLICATION_NAME
--deployment-group-name DEPLOYMENT_GROUP_NAME --
revision "revisionType=S3,s3Location={bucket=my-revision-
bucket,bundleType=zip,key=httpd_app.zip}"
```

Comme il existe désormais une InService instance dans le groupe Auto Scaling, ce déploiement devrait fonctionner et le message d'erreur « Le déploiement a échoué car aucune instance n'a été trouvée pour votre groupe de déploiement » ne devrait plus s'afficher.

8. Une fois le déploiement réussi, redimensionnez votre groupe Auto Scaling à sa capacité initiale, si vous l'avez déjà redimensionné :

```
aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME
--desired-capacity ORIGINAL_CAPACITY
```

Pour résoudre le problème de blocage du déploiement (console)

1. (Facultatif) Bloquez les CI/CD pipelines à l'origine de l' CodeDeploy erreur afin d'éviter des déploiements inattendus pendant que vous résolvez ce problème.
2. Accédez à la console Amazon EC2 et prenez note de votre paramètre de capacité Auto Scaling Desired. Vous devrez peut-être revenir à ce chiffre à la fin de cette procédure. Pour plus d'informations sur la recherche de ce paramètre, consultez la section [Définition des limites de capacité sur votre groupe Auto Scaling](#).
3. Définissez le nombre d'instances EC2 souhaité comme suit : 1

Ceci est facultatif si la capacité souhaitée était supérieure 1 à la capacité initiale. En le réduisant à 1, l'instance mettra moins de temps à provisionner et à déployer ultérieurement, ce qui accélère

le dépannage. Si votre capacité Auto Scaling Desired était initialement définie sur 0, vous devez l'augmenter à 1. Cela est obligatoire.

 Note

Les étapes restantes de cette procédure supposent que vous avez défini la capacité souhaitée sur 1.

- a. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>, puis sélectionnez Auto Scaling Groups dans le volet de navigation.
 - b. Choisissez la région appropriée.
 - c. Accédez au groupe Auto Scaling problématique.
 - d. Dans Détails du groupe, choisissez Modifier.
 - e. Réglez la capacité souhaitée sur 1.
 - f. Choisissez Mettre à jour.
4. Désenregistrez le groupe Auto Scaling du groupe de déploiement :

 Warning

Les sous-étapes suivantes lanceront une nouvelle instance EC2 sans logiciel. Avant d'exécuter la commande, assurez-vous qu'une InService instance Auto Scaling n'exécutant aucun logiciel est acceptable. Par exemple, assurez-vous que l'équilibreur de charge associé à l'instance n'envoie pas de trafic vers cet hôte sans logiciel.

- a. Ouvrez la CodeDeploy console à l'adresse <https://console.aws.amazon.com/codedeploy/>.
- b. Choisissez la région appropriée.
- c. Dans le volet de navigation, choisissez Applications.
- d. Choisissez le nom de votre CodeDeploy application.
- e. Choisissez le nom de votre groupe CodeDeploy de déploiement.
- f. Choisissez Modifier.
- g. Dans Configuration de l'environnement, désélectionnez les groupes Amazon EC2 Auto Scaling.

 Note

La console ne vous permet pas d'enregistrer la configuration si aucune configuration d'environnement n'est définie. Pour contourner cette vérification, ajoutez temporairement un tag de EC2 ou On-premises dont vous savez qu'il ne sera résolu à aucun hôte. Pour ajouter une balise, sélectionnez des instances Amazon EC2 ou une instance sur site, puis ajoutez une balise Key of or. **EC2 On-premises**. Vous pouvez laisser le tag Value vide.

- h. Sélectionnez Enregistrer les modifications.

Une fois ces sous-étapes terminées, voici ce qui se produit :

- i. CodeDeploy désenregistre le groupe Auto Scaling du groupe de déploiement.
- ii. CodeDeploy supprime le hook du cycle de vie Auto Scaling du groupe Auto Scaling.
- iii. Le hook à l'origine de l'échec du déploiement n'étant plus présent, Auto Scaling annule l'instance EC2 existante et en lance immédiatement une nouvelle pour l'adapter à la capacité souhaitée. La nouvelle instance devrait bientôt passer à InService l'état. La nouvelle instance n'inclut aucun logiciel.

- 5. Attendez que l'instance EC2 entre dans l'InService état. Pour vérifier son état :

- a. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
- b. Dans le panneau de navigation, choisissez Groupes Auto Scaling.
- c. Choisissez votre groupe Auto Scaling.
- d. Dans le volet de contenu, choisissez l'onglet Instance Management.
- e. Sous Instances, assurez-vous que la colonne Lifecycle est indiquée à InService côté de l'instance.

- 6. Réenregistrez le groupe Auto Scaling auprès du groupe de CodeDeploy déploiement en utilisant la même méthode que celle que vous avez utilisée pour le supprimer :

- a. Ouvrez la CodeDeploy console à l'adresse <https://console.aws.amazon.com/codedeploy/>.
- b. Choisissez la région appropriée.
- c. Dans le volet de navigation, choisissez Applications.
- d. Choisissez le nom de votre CodeDeploy application.
- e. Choisissez le nom de votre groupe CodeDeploy de déploiement.

- f. Choisissez Modifier.
- g. Dans Configuration de l'environnement, sélectionnez les groupes Amazon EC2 Auto Scaling et sélectionnez votre groupe Auto Scaling dans la liste.
- h. Sous Instances Amazon EC2 ou Instances sur site, recherchez le tag que vous avez ajouté et supprimez-le.
- i. Décochez la case à côté des instances Amazon EC2 ou des instances sur site.
- j. Sélectionnez Enregistrer les modifications.

Cette configuration réinstalle le hook du cycle de vie dans le groupe Auto Scaling.

7. Créez un déploiement à l'échelle du parc à l'aide de l'Amazon S3 ou de GitHub la version dont vous savez qu'elle est saine et que vous souhaitez utiliser.

Par exemple, si la révision est un fichier .zip dans un compartiment Amazon S3 appelé my-revision-bucket avec une clé d'objet de `httpd_app.zip`, procédez comme suit :

- a. Dans la CodeDeploy console, sur la page Groupe de déploiement, choisissez Créer un déploiement.
- b. Pour Type de révision, choisissez Mon application est stockée dans Amazon S3.
- c. Pour Emplacement de la révision, sélectionnez `s3://my-revision-bucket/httpd_app.zip`.
- d. Pour le type de fichier de révision, sélectionnez .zip.
- e. Choisissez Créer un déploiement.

Comme il existe désormais une InService instance dans le groupe Auto Scaling, ce déploiement devrait fonctionner et le message d'erreur « Le déploiement a échoué car aucune instance n'a été trouvée pour votre groupe de déploiement » ne devrait plus s'afficher.

8. Une fois le déploiement réussi, redimensionnez votre groupe Auto Scaling à sa capacité initiale, si vous l'avez déjà redimensionné :
 - a. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>, puis sélectionnez Auto Scaling Groups dans le volet de navigation.
 - b. Choisissez la région appropriée.
 - c. Accédez à votre groupe Auto Scaling.
 - d. Dans Détails du groupe, choisissez Modifier.

- e. Rétablissez la capacité souhaitée à sa valeur initiale.
- f. Choisissez Mettre à jour.

Codes d'erreur pour AWS CodeDeploy

Cette rubrique fournit des informations de référence sur CodeDeploy les erreurs.

Code d'erreur	Description
AGENT_ISSUE	Le déploiement a échoué en raison d'un problème lié à l' CodeDeploy agent. Assurez-vous que l'agent est installé et en cours d'exécution sur toutes les instances de ce groupe de déploiement. En savoir plus : • Vérifiez que l' CodeDeploy agent est en cours d'exécution • Installation de l' CodeDeploy agent • Travailler avec l' CodeDeploy agent
AUTO_SCALING_IAM_ROLE_PERMISSIONS	Le rôle de service associé à votre groupe de déploiement ne dispose pas de l'autorisation requise pour effectuer des opérations dans le AWS service suivant. En savoir plus : • Étape 2 : créer un rôle de service pour CodeDeploy • Création d'un rôle pour déléguer des autorisations à un AWS service
HEALTH_CONSTRAINTS	Le déploiement global a échoué, car trop d'instances individuelles n'ont pas réussi le déploiement, trop peu d'instances saines sont

Code d'erreur	Description
	disponibles pour le déploiement ou certaines instances de votre groupe de déploiement rencontrent des problèmes. En savoir plus : • Instance Health • Résolution des problèmes d'instance • Résoudre les problèmes de déploiement d'EC2/sur site
HEALTH_CONSTRAINTS_INVALID	Le déploiement ne peut pas démarrer, car le nombre minimal d'instances saines, tel que défini par la configuration de votre déploiement, n'est pas disponible. Vous pouvez réduire le nombre requis d'instances saines en mettant à jour votre configuration de déploiement ou augmenter le nombre d'instances dans ce groupe de déploiement. En savoir plus : • Instance Health • Utilisation d'instances pour CodeDeploy

Code d'erreur	Description
IAM_ROLE_MISSING	Le déploiement a échoué car il n'existe aucun rôle de service avec le nom du rôle de service spécifié pour le groupe de déploiement. Assurez-vous d'utiliser le nom de rôle de service correct. En savoir plus : • Étape 2 : créer un rôle de service pour CodeDeploy • Modifiez les paramètres du groupe de déploiement avec CodeDeploy
IAM_ROLE_PERMISSIONS	CodeDeploy ne dispose pas des autorisations requises pour assumer un rôle, ou le rôle IAM que vous utilisez ne vous autorise pas à effectuer des opérations dans un AWS service. En savoir plus : • Étape 1 : Configuration • Étape 2 : créer un rôle de service pour CodeDeploy • Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2

Code d'erreur	Description
NO_INSTANCES	Cela peut être dû à l'une des raisons suivantes. • Pour un déploiement EC2 bleu/vert /On-Premises, si vous utilisez des EC2 balises Amazon, elles risquent de ne pas être configurées correctement. Dans votre groupe de CodeDeploy déploiement, assurez-vous qu'ils sont inclus dans vos instances bleues et vos instances vertes. Vous pouvez vérifier que vos instances sont correctement étiquetées à l'aide de la EC2 console Amazon. • Si vous utilisez un groupe Amazon EC2 Auto Scaling, il se peut que votre groupe Auto Scaling ne dispose pas d'une capacité suffisante. Assurez-vous que votre groupe Auto Scaling dispose d'une capacité suffisante pour votre déploiement. Vous pouvez consulter la capacité de votre groupe Amazon EC2 Auto Scaling en examinant son nombre d'instances saines à l'aide de la EC2 console Amazon. • Pour un déploiement EC2 bleu/vert /On-Premises, les flottes bleues et vertes peuvent ne pas être de la même taille. Assurez-vous que les deux flottes sont de la même taille. En savoir plus : • Tagging Instances for Deployments • Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling • Création d'une application pour un blue/green déploiement (console)

Code d'erreur	Description
OVER_MAX_INSTANCES	Le déploiement a échoué, car plus d'instances sont ciblées pour le déploiement que d'instances sont autorisées pour votre compte. Afin de réduire le nombre d'instances ciblées pour ce déploiement, mettez à jour les paramètres de balise pour ce groupe de déploiement ou supprimez certaines des instances ciblées. Vous pouvez également nous contacter AWS Support pour demander une augmentation de limite. En savoir plus : • Modifiez les paramètres du groupe de déploiement avec CodeDeploy • CodeDeploy quotas • Demander une augmentation de limite.
THROTTLED	Le déploiement a échoué car le nombre de demandes effectuées est supérieur à celui autorisé AWS CodeDeploy par un rôle IAM. Essayez de réduire le nombre de demandes. En savoir plus : • Débit de demandes d'API de requête

Code d'erreur	Description
UNABLE_TO_SEND_ASG	Le déploiement a échoué car le groupe de déploiement n'est pas correctement configuré avec son groupe Amazon EC2 Auto Scaling. Dans la CodeDeploy console, supprimez le groupe Amazon EC2 Auto Scaling du groupe de déploiement, puis ajoutez-le à nouveau. En savoir plus : • Intégration entre Under the Hood CodeDeploy et Auto Scaling

Rubriques en relation

[Résolution des problèmes CodeDeploy](#)

CodeDeploy ressources

Les ressources connexes suivantes peuvent vous aider dans votre travail avec CodeDeploy.

Guides de référence et ressources d'assistance

- [AWS CodeDeploy Référence d'API](#) : descriptions, syntaxes et exemples d'utilisation concernant CodeDeploy les actions et les types de données, y compris les paramètres courants et les codes d'erreur.
- [CodeDeploy technique FAQs](#) — Les questions les plus fréquemment posées par les clients à propos de CodeDeploy.
- [AWS centre d'assistance](#) — Le centre de création et de gestion de vos AWS Support dossiers. Comprend également des liens vers d'autres ressources, telles que des forums, des informations techniques FAQs, l'état de santé des services et AWS Trusted Advisor.
- [AWS plans de support](#) : page Web principale contenant des informations sur AWS Support les plans.
- [Contactez-nous](#) — Un point de contact central pour les demandes concernant la AWS facturation, le compte, les événements, les abus et autres problèmes.
- [AWS conditions du site](#) — Informations détaillées sur nos droits d'auteur et notre marque commerciale ; votre compte, votre licence et l'accès au site ; et d'autres sujets.

Exemples

- [CodeDeploy samples on GitHub](#) — Exemples et modèles de scénarios pour CodeDeploy.
- CodeDeploy Plugin [Jenkins — Plugin](#) Jenkins pour. CodeDeploy
- [CodeDeploy agent](#) : version open source de l' CodeDeploy agent.

Blogs

- [AWS DevOps blog](#) — Informations pour les développeurs, les administrateurs système et les architectes.

AWS kits et outils de développement logiciel

Les outils AWS SDKs et outils suivants permettent de développer des solutions avec CodeDeploy :

- [AWS SDK pour .NET](#)
- [AWS SDK pour Java](#)
- [AWS SDK pour JavaScript](#)
- [AWS SDK pour PHP](#)
- [AWS SDK pour Python \(Boto\)](#)
- [AWS SDK pour Ruby](#)
- AWS Toolkit for Eclipse — Parties [1](#), [2](#) et [3](#).
- [AWS Tools for Windows PowerShell](#)— Ensemble d' PowerShell applets de commande Windows qui présentent les fonctionnalités de AWS SDK pour .NET l' PowerShell environnement.
- [CodeDeploy applets de commande dans le Outils AWS pour PowerShell](#) — Ensemble d' PowerShell applets de commande Windows qui exposent les fonctionnalités de CodeDeploy l'environnement. PowerShell
- [AWS Command Line Interface](#)— Syntaxe de ligne de commande uniforme pour accéder aux AWS services. AWS CLI utilise un processus de configuration unique pour permettre l'accès à tous les services pris en charge.
- [AWS outils](#) de développement : liens vers des outils de développement et des ressources fournissant de la documentation, des exemples de code, des notes de publication et d'autres informations pour vous aider à créer des applications innovantes avec CodeDeploy et AWS.

Historique du document

Le tableau suivant décrit les principales modifications apportées à ce guide de l'utilisateur pour prendre en charge les fonctionnalités nouvelles et améliorées depuis la dernière version du Guide de l'utilisateur CodeDeploy .

- Version de l'API : 06/10/2014

Modification	Description	Date
CodeDeploy version de l'agent v1.8.0	L' AWS CodeDeploy agent a été mis à jour vers la version 1.8.0. Pour plus d'informations, consultez la section Historique des versions de l' CodeDeploy agent .	31 juillet 2025
CodeDeploy a mis à jour une politique AWS gérée existante	AWSCodeDeployDeployerAccess Il a été mis à jour. Pour plus d'informations, consultez la section Mises à jour des politiques AWS gérées .	16 décembre 2024
CodeDeploy a mis à jour une politique AWS gérée existante	AWSCodeDeployReadOnlyAccess Il a été mis à jour. Pour plus d'informations, consultez la section Mises à jour des politiques AWS gérées .	16 décembre 2024
CodeDeploy a mis à jour une politique AWS gérée existante	AWSCodeDeployFullAccess Il a été mis à jour. Pour plus d'informations, consultez la section Mises	16 décembre 2024

[à jour des politiques AWS
gérées.](#)

[CodeDeploy version de l'agent
v1.7.1](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.7.1. Pour plus d'informations, consultez [la section Historique des versions de l' CodeDeploy agent](#).

14 novembre 2024

[Noms des compartiments
Amazon S3 mis à jour](#)

Les exemples de compartiments Amazon S3 présentés dans ce guide ont été mis à jour afin d'utiliser les noms réservés par AWS.

17 juin 2024

[Ajout d'un texte alternatif
\(texte alternatif\)](#)

Toutes les images de ce guide ont été mises à jour pour inclure du texte alternatif. Le texte alternatif est lu par les lecteurs d'écran et rend notre documentation plus accessible aux utilisateurs aveugles.

22 mai 2024

[CodeDeploy version de l'agent
v1.7.0](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.7.0. Pour plus d'informations, consultez [la section Historique des versions de l' CodeDeploy agent](#).

6 mars 2024

Commandes modifiées

Les `sudo service codedeploy-agent status/start/stop` commandes ne sont plus recommandées car elles ne sont pas utilisées `systemd` pour la gestion des processus des CodeDeploy agents, ce qui est une bonne pratique. Pour vous assurer `systemd` qu'il est utilisé, utilisez des `systemctl` commandes, comme indiqué dans l'exemple suivant : `systemctl start codedeploy-agent` .

Les rubriques suivantes ont été mises à jour avec des `systemctl` commandes : [installation de l' CodeDeploy agent pour Amazon Linux ou RHEL](#), [installation de l' CodeDeploy agent pour Ubuntu Server](#), [résolution de tous les événements du cycle de vie](#), [erreurs ignorées](#) et [création d'un nouveau fichier CodeDeploy journal s'il a été supprimé accidentellement](#).

12 janvier 2024

Sujets ajoutés

Les rubriques [Gestion du processus d' CodeDeploy agent](#) et [Référencement des fichiers](#) ont été ajoutées dans [les scripts d'événements de votre cycle de vie](#).

12 janvier 2024

[CodeDeploy prend désormais en charge les configurations zonales](#)

Mise à jour de la CodeDeploy rubrique [Créer une configuration de déploiement](#) avec des informations de configuration zonale.

7 décembre 2023

[CodeDeploy prend désormais en charge les déploiements de terminaison](#)

La rubrique [Enabling termination deployments during Auto Scale-in events](#) a été ajoutée pour décrire la fonctionnalité de déploiement de terminaison. Nous avons également mis à jour la [section « hooks AppSpec »](#) pour un déploiement EC2/sur site, les [rubriques Créer un groupe de déploiement pour un déploiement sur place \(console\)](#) et [Créer un groupe de déploiement pour un déploiement EC2/sur site \(console\)](#) afin de tenir [blue/green compte](#) de cette fonctionnalité.

7 décembre 2023

[Formatage JSON fixe](#)

Correction du formatage des exemples de code JSON dans la [rubrique « Ressources AppSpec »](#) (Amazon ECS et AWS Lambda déploiements uniquement).

3 décembre 2023

[Ajout d'une rubrique sur la résolution des problèmes](#)

Ajout d'une rubrique relative à la [résolution des problèmes de déploiement d'Amazon ECS](#).

24 octobre 2023

Nom du AppSpec fichier mis à jour	Mise à jour de la référence du CodeDeploy AppSpec fichier pour indiquer que le AppSpec fichier doit être nommé <code>appspec.yml</code> pour les déploiements EC2/sur site.	5 octobre 2023
CodeDeploy prend désormais en charge plusieurs équilibreurs de charge	Les rubriques Créer un groupe de déploiement pour un déploiement sur place (console) , Créer un groupe de déploiement pour un déploiement EC2/sur site blue/green (console) et Configurer un équilibreur de charge ont été mises à jour dans Elastic Load Balancing pour les déploiements Amazon EC2 CodeDeploy afin d'indiquer la prise en charge de plusieurs équilibreurs de charge.	26 septembre 2023
Mise à jour des régions dans la rubrique VPC	Le tableau de la rubrique Utilisation CodeDeploy avec Amazon Virtual Private Cloud a été mis à jour pour indiquer le support régional supplémentaire. Plus précisément, les régions Asie-Pacifique (Hyderabad), Asie-Pacifique (Melbourne), Europe (Milan), Europe (Espagne) et Europe (Zurich) ont été mises à jour pour indiquer la prise en charge du point de terminaison de l'agent.	22 septembre 2023

[Régions mises à jour dans la rubrique du kit de ressources](#)

Les AWS régions suivantes ont été ajoutées aux [noms de compartiments du kit de ressources par](#) section régionale : Asie-Pacifique (Osaka), Asie-Pacifique (Hyderabad), Canada (centre), Europe (Espagne), Europe (Zurich), Moyen-Orient (Émirats arabes unis). Les politiques IAM ont également été mises à jour avec ces régions et toutes les autres régions manquantes.

22 septembre 2023

[Rubriques d'installation et de mise à jour des agents abrégées](#)

Abréviation des rubriques [Installer l' CodeDeploy agent pour Windows Server](#) et [Mettre à jour l' CodeDeploy agent sur Windows Server](#).
Suppression du compartiment Amazon S3 redondant URLs et des commandes de copie Amazon S3.

21 septembre 2023

[Ajout de la région Asie-Pacifique \(Jakarta\)](#)

La région Asie-Pacifique (Jakarta) a été ajoutée aux [noms de compartiments du kit de ressources par région](#).

21 septembre 2023

[CodeDeploy a mis à jour une politique AWS gérée existante](#)

La politique AWSCodeDeployRole gérée a été mise à jour. Pour plus d'informations, consultez la rubrique [Mises à jour AWS des politiques gérées par AWS](#).

16 août 2023

Ajout d'une limite	Ajout d'une limite à la rubrique CodeDeploy des limites . La limite est le nombre maximum d'alarmes associées à un groupe de déploiement.	15 août 2023
Correction d'une étape liée aux équilibres de charge	Correction des instructions dans Créer un groupe de déploiement pour un blue/green déploiement EC2/sur site (console) . L'étape de l'équilibreur de charge est désormais marquée comme facultative.	3 août 2023
Libellé clarifié dans la rubrique Amazon ECS	Clarification du libellé du didacticiel : Déployer une application dans Amazon ECS . Le libellé indique désormais que vous déployez une application. Auparavant, le libellé indiquait que vous déployiez un service Amazon ECS.	3 août 2023
CodeDeploy est désormais disponible dans la région Israël (Tel Aviv)	CodeDeploy est désormais disponible dans la région d'Israël (Tel Aviv) (il-central-1). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	31 juillet 2023

Mise à jour du sujet	Mise à jour de la rubrique Résoudre les problèmes de déploiement d'EC2/sur site avec un conseil sur l'utilisation d'un runbook pour effectuer automatiquement des tâches de dépannage.	7 juillet 2023
Mise à jour du sujet	Mise à jour de la section « ressources AppSpec » pour les déploiements Amazon ECS avec plus d'informations sur l'ARN de définition des tâches.	7 juillet 2023
Mise à jour du sujet	Mise à jour de la rubrique Étape 1 : Installation et configuration de l'instance AWS CLI sur site avec des informations de dépannage.	7 juillet 2023
Mise à jour du sujet	Mise à jour de la rubrique sur la prévention de la confusion entre services adjoints avec des informations sur les blue/green déploiements d'Amazon ECS via. CloudFormation	6 juillet 2023
Mise à jour du sujet	La rubrique sur la prévention de la confusion entre les services adjoints a été mise à jour avec des informations sur les blue/green déploiements d'Amazon ECS via. CloudFormation	6 juillet 2023

Mise à jour du sujet	Mise à jour de la rubrique Configurations de déploiement prédéfinies pour une plateforme de calcul EC2/sur site . Ajout d'une note concernant le comportement de la configuration de déploiement CodeDeployDefault. HalfAtATime prédéfinie avec les groupes Auto Scaling.	29 juin 2023
Mise à jour du sujet	Mise à jour de la AWS CodeDeploy rubrique Sécurité de l'infrastructure pour indiquer les nouvelles versions minimales et recommandées du protocole TLS (Transport Layer Security).	28 juin 2023
Mise à jour des limites	La limite suivante a été modifiée : « Nombre maximum d'heures pendant lequel un déploiement EC2/sur site peut être exécuté ». Pour plus d'informations, consultez les Limites	27 juin 2023
Mise à jour du sujet	La rubrique Étape 3 : Limiter les autorisations de CodeDeploy l'utilisateur a été mise à jour avec des instructions détaillées.	31 mai 2023

CodeDeploy a mis à jour une politique AWS gérée existante	La politique AWSCodeDeployFullAccess gérée a été mise à jour. Pour plus d'informations, consultez la rubrique Mises à jour AWS des politiques gérées par AWS .	16 mai 2023
CodeDeploy version de l'agent v1.6.0	L' AWS CodeDeploy agent a été mis à jour vers la version 1.6.0. Pour plus d'informations, consultez la section Historique des versions de l' CodeDeploy agent .	30 mars 2023
CodeDeploy version de l'agent v1.5.0	L' AWS CodeDeploy agent a été mis à jour vers la version 1.5.0. Pour plus d'informations, consultez la section Historique des versions de l' CodeDeploy agent .	3 mars 2023
Mise à jour de la plateforme de calcul Amazon ECS	Les déploiements sur une plate-forme de calcul Amazon ECS sont désormais pris en charge dans la région Asie-Pacifique (Jakarta).	8 février 2023
CodeDeploy a mis à jour une politique AWS gérée existante	La politique AWSCodeDeployRole gérée a été mise à jour. Pour plus d'informations, consultez la rubrique Mises à jour AWS des politiques gérées par AWS .	3 février 2023

Mise à jour du sujet	La rubrique Utilisation CodeDeploy avec Amazon Virtual Private Cloud a été mise à jour avec les AWS régions nouvelles et modifiées.	2 février 2023
Mises à jour thématiques	CodeDeploy est désormais disponible dans la région Asie-Pacifique (Melbourne) (ap-southeast-4). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de ces nouvelles régions.	26 janvier 2023
Mises à jour des meilleures pratiques de sécurité	La CodeDeploy section Mise en route et quelques autres sections ont été mises à jour pour être conformes aux meilleures pratiques en matière AWS de sécurité.	23 janvier 2023
CodeDeploy version de l'agent v1.4.1	L' AWS CodeDeploy agent a été mis à jour vers la version 1.4.1. Pour plus d'informations, consultez la section Historique des versions de l' CodeDeploy agent .	6 décembre 2022

[Ajout d'une rubrique sur la résolution des problèmes](#)

Ajout d'une rubrique expliquant comment résoudre les erreurs causées par de longs chemins de fichiers utilisés avec l' AWS CodeDeploy agent pour Windows. Pour plus d'informations, voir [Les chemins de fichier longs provoquent des erreurs « Aucun fichier ou répertoire de ce type »](#).

6 décembre 2022

[Modification d'une limite](#)

La limite suivante a été modifiée : « Nombre maximum de configurations de déploiement personnalisées associées à un AWS compte ». La limite est désormais de 200. Pour plus d'informations sur les limites, consultez la rubrique [Limites](#).

7 septembre 2022

[CodeDeploy version de l'agent v1.4.0](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.4.0. Pour plus d'informations, consultez [la section Historique des versions de l' AWS CodeDeploy agent](#).

31 août 2022

Quelques limites ont été fixées.

Les limites suivantes ont été fixées : « Le nombre maximum de déploiements simultanés associés à un AWS compte » est désormais de 1 000. Le « nombre maximum d'instances dans un seul déploiement » est désormais de 1 000. « Le nombre maximum d'instances pouvant être utilisées par des déploiements simultanés en cours et associés à un compte » est désormais de 1 000. « Le nombre maximum de configurations de déploiement personnalisées associées à un AWS compte » est désormais de 100. Pour plus d'informations sur les limites, consultez la rubrique [Limites](#).

8 août 2022

Ajout d'un tableau indiquant les CodeDeploy points de terminaison pris en charge dans chaque région.

Pour plus d'informations, consultez [Utiliser CodeDeploy avec Amazon Virtual Private Cloud](#).

20 avril 2022

Ajout d'une nouvelle limite pour les blue/green déploiements Amazon ECS.	Le nombre maximum d'heures entre le déploiement d'une révision et le transfert du trafic vers l'environnement de remplacement lors d'un déploiement bleu/vert d'Amazon ECS est désormais de 120. Pour plus d'informations, consultez la section Déploiements dans la rubrique Limites .	12 avril 2022
Ajout d'un sujet sur la façon de prévenir le problème de confusion des députés	Pour plus d'informations, consultez Gestion des identités et des accès AWS pour AWS CodeDeploy .	14 mars 2022
CodeDeploy a mis à jour une politique AWS gérée existante	Le AmazonEC2RoleforAWSCodeDeployLimited rôle a été mis à jour. Pour plus d'informations, consultez la section Mises à jour des politiques AWS gérées .	22 novembre 2021
CodeDeploy a mis à jour une politique AWS gérée existante	AWS CodeDeployRole II a été mis à jour. Pour plus d'informations, consultez la section Mises à jour des politiques AWS gérées .	18 mai 2021
CodeDeploy version de l'agent v1.3.2	L' AWS CodeDeploy agent a été mis à jour vers la version 1.3.2. Pour plus d'informations, consultez l'historique des versions de l' CodeDeploy agent .	6 mai 2021

[CodeDeploy prend en charge la mise à jour des instances Amazon EC2 obsolètes](#)

CodeDeploy prend désormais en charge la mise à jour automatique des instances Amazon EC2 obsolètes. Pour plus d'informations, voir [Configurer les options avancées pour un groupe de déploiement](#).

23 février 2021

[CodeDeploy version de l'agent v1.3.1](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.3.1. Pour plus d'informations, consultez [l'historique des versions de l' CodeDeploy agent](#).

22 décembre 2020

[CodeDeploy version de l'agent v1.3.0](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.3.0. Pour plus d'informations, consultez [l'historique des versions de l' CodeDeploy agent](#).

10 novembre 2020

[CodeDeploy version de l'agent v1.2.1](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.2.1. Pour plus d'informations, consultez [l'historique des versions de l' CodeDeploy agent](#).

23 septembre 2020

[CodeDeploy prend en charge les points de terminaison Amazon VPC alimentés par AWS PrivateLink](#)

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos AWS ressources, vous pouvez établir une connexion privée entre votre VPC et CodeDeploy. Vous pouvez utiliser cette connexion pour permettre à CodeDeploy de communiquer avec vos ressources sur votre VPC sans passer par le réseau Internet public. Pour plus d'informations, consultez [Utiliser CodeDeploy avec Amazon Virtual Private Cloud](#).

11 août 2020

[Limites CodeDeploy de service mises à jour](#)

Mise à jour de la limite du nombre d'applications par compte et du nombre de groupes de déploiement par application à 1 000. Pour plus d'informations sur les limites CodeDeploy de service, consultez la section [CodeDeploy Limites](#).

6 août 2020

[CodeDeploy version de l'agent v1.1.2](#)

L' AWS CodeDeploy agent a été mis à jour vers la version 1.1.2. Pour plus d'informations, consultez [l'historique des versions de l' CodeDeploy agent](#).

4 août 2020

[CodeDeploy sortie de l'agent
1.1.0 et intégration avec
Amazon EC2 Systems
Manager](#)

La version 1.1.0 de l'agent CodeDeploy est désormais disponible. Pour plus d'informations, consultez [l'historique des versions de l'agent CodeDeploy](#).

30 juin 2020

Vous pouvez désormais utiliser Amazon EC2 Systems Manager pour gérer automatiquement l'installation et les mises à jour de CodeDeploy sur votre agent sur vos instances Amazon EC2 ou sur site. Pour plus d'informations, consultez [Installer l'agent CodeDeploy à l'aide d'Amazon EC2 Systems Manager](#).

[CodeDeploy prend en charge la gestion des déploiements Amazon ECS avec CloudFormation](#)

Vous pouvez désormais utiliser CloudFormation pour gérer les déploiements Amazon ECS via CodeDeploy. Vous générez votre déploiement en définissant vos ressources vertes et bleues et en spécifiant les paramètres de routage et de stabilisation du trafic à utiliser dans CloudFormation. Pour plus d'informations, consultez [Créer un déploiement bleu/vert Amazon ECS](#) via CloudFormation.

19 mai 2020

[CodeDeploy prend en charge le transfert pondéré du trafic pour les blue/green déploiements Amazon ECS](#)

CodeDeploy prend désormais en charge le transfert pondéré du trafic pour les déploiements bleu/vert d'Amazon ECS. Vous choisissez ou créez une configuration de déploiement pour spécifier le nombre d'intervalles de déplacement de trafic dans le déploiement et le pourcentage de trafic à déplacer dans chaque intervalle. La rubrique suivante a été mise à jour pour refléter cette modification : [Configurations de déploiement sur une plateforme de calcul Amazon ECS](#).

6 février 2020

[Rubriques mises à jour sur la sécurité, l'authentification et le contrôle d'accès](#)

Les informations relatives à la sécurité, à l'authentification et au contrôle d'accès ont été organisées dans un nouveau chapitre sur la sécurité. Pour plus d'informations, consultez [Sécurité](#).

26 novembre 2019

[CodeDeploy prend en charge les règles de notification](#)

Vous pouvez désormais utiliser des règles de notification pour informer les utilisateurs des modifications importantes apportées aux déploiements. Pour de plus amples informations, veuillez consulter [Création d'une règle de notification](#).

5 novembre 2019

Sujets mis à jour

CodeDeploy est désormais disponible dans la région Asie-Pacifique (Hong Kong) (ap-east-1). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région. Vous devez activer explicitement l'accès à cette région. Pour plus d'informations, consultez la section [Gestion des AWS régions](#).

25 avril 2019

Sujets mis à jour

AWS CodeDeploy prend désormais en charge blue/green les déploiements d'une application conteneurisée dans un service Amazon ECS. Une CodeDeploy application qui utilise la nouvelle plateforme de calcul Amazon ECS déploie une application conteneurisée vers un nouvel ensemble de tâches de remplacement dans le même service Amazon ECS. Plusieurs rubriques ont été ajoutées et mises à jour pour refléter ce changement, notamment la [présentation des plateformes de AWS CodeDeploy calcul](#), [les déploiements sur une plateforme de calcul Amazon ECS](#), la [structure de AppSpec fichiers pour les déploiements Amazon ECS](#) et la [création d'une application pour le déploiement d'un service Amazon ECS \(console\)](#).

27 novembre 2018

CodeDeploy Agent mis à jour

L' AWS CodeDeploy agent a été mis à jour vers la version 1.0.1.1597. Pour plus d'informations, consultez [la section Historique des versions de l' CodeDeploy agent](#).

15 novembre 2018

Console mise à jour	Les procédures de ce guide ont été mises à jour pour correspondre à la nouvelle version de la console CodeDeploy.	30 octobre 2018
Nouvelle version minimale prise en charge de l' CodeDeploy agent	La version minimale prise en charge de l' AWS CodeDeploy agent est désormais 1.7.x. Pour plus d'informations, consultez la section Historique des versions de l' CodeDeploy agent .	7 août 2018

Mises à jour antérieures

Le tableau ci-après décrit des modifications importantes apportées dans chaque version du Guide de l'utilisateur AWS CodeDeploy avant juin 2018.

Modifier	Description	Date de modification
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Europe (Paris) (eu-west-3). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	19 décembre 2017
Mise à jour des rubriques	CodeDeploy est désormais disponible dans la région Chine (Ningxia). Pour utiliser les services de la région Chine (Pékin) ou de la région Chine (Ningxia), vous devez disposer d'un compte et d'informations d'identification pour ces régions. Les comptes et informations d'identification AWS des autres régions ne fonctionnent pas pour les régions de Pékin et du Ningxia, et vice versa.	11 décembre 2017

Modifier	Description	Date de modification
	Les informations relatives à certaines ressources pour les régions de Chine, telles que les noms des compartiments du kit de CodeDeploy ressources et les procédures d'installation des CodeDeploy agents, ne sont pas incluses dans cette édition du guide de CodeDeploy l'utilisateur. Pour plus d'informations, consultez: • CodeDeploy dans Commencer AWS dans la région de Chine (Pékin) • CodeDeploy Guide de l'utilisateur pour les régions de Chine (version anglaise version chinoise)	
Mise à jour des rubriques	CodeDeploy prend désormais en charge le déploiement d'une fonction Lambda. Un déploiement AWS Lambda déplace le trafic entrant d'une fonction Lambda existante vers une version de fonction Lambda mise à jour. Vous choisissez ou créez une configuration de déploiement pour spécifier le nombre d'intervalles de transfert de trafic dans le déploiement et le pourcentage de trafic à déplacer dans chaque intervalle. AWS Lambda les déploiements sont pris en charge par le modèle d'application AWS sans serveur (AWS SAM) afin que vous puissiez utiliser une préférence de déploiement AWS SAM pour gérer la manière dont le trafic est transféré au cours d'un déploiement. AWS Lambda Plusieurs rubriques ont été ajoutées ou mises à jour afin de tenir compte de cette modification, notamment Vue d'ensemble des plateformes CodeDeploy informati ques, Déploiements sur une plateforme de calcul AWS Lambda, Création d'un déploiement de plateforme de calcul AWS Lambda (console), Création d'une application pour le déploiement d'une AWS Lambda fonction (console) et Ajouter un AppSpec fichier pour un déploiement AWS Lambda.	28 novembre 2017

Modifier	Description	Date de modification
Nouvelle rubrique	CodeDeploy prend désormais en charge les déploiements directement sur une machine ou une instance locale sur laquelle l' CodeDeploy agent est installé. Vous pouvez tester localement un déploiement et, s'il comporte des erreurs, utiliser les journaux d'erreurs de l' CodeDeploy agent pour le déboguer. Vous pouvez également utiliser un déploiement local pour tester l'intégrité d'une révision d'application, le contenu d'un AppSpec fichier, etc. Pour de plus amples informations, veuillez consulter Utiliser l' CodeDeploy agent pour valider un package de déploiement sur une machine locale .	16 novembre 2017
Mise à jour des rubriques	CodeDeploy la prise en charge des équilibres de charge Elastic Load Balancing dans les groupes de déploiement a été étendue pour inclure les équilibres de charge réseau pour les blue/green déploiements et les déploiements sur place. Vous pouvez désormais choisir un Application Load Balancer, un Classic Load Balancer ou un Network Load Balancer pour votre groupe de déploiement. Les équilibres de charge sont obligatoires pour les blue/green déploiements et facultatifs pour les déploiements sur place. Des rubriques ont été mises à jour pour refléter cette nouvelle prise en charge, notamment Integrating CodeDeploy with Elastic Load Balancing , Création d'une application pour un déploiement sur place (console) , Conditions préalables au déploiement , Integrating CodeDeploy with Elastic Load Balancing et Création d'une application pour un déploiement sur place (console) .	12 septembre 2017

Modifier	Description	Date de modification
Mise à jour des rubriques	CodeDeploy la prise en charge des équilibreurs de charge Elastic Load Balancing dans les groupes de déploiement a été étendue pour inclure les équilibreurs de charge d'application pour les blue/green déploiements et les déploiements sur place. Vous pouvez désormais choisir entre un Application Load Balancer et un Classic Load Balancer pour votre groupe de déploiement. Les équilibreurs de charge sont obligatoires pour les blue/green déploiements et facultatifs pour les déploiements sur place. Les rubriques Integrating CodeDeploy with Elastic Load Balancing , Créez une application avec CodeDeploy et Créez un groupe de déploiement avec CodeDeploy ont été mises à jour pour refléter cette nouvelle prise en charge.	le 10 août 2017

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	CodeDeploy prend désormais en charge l'utilisation de plusieurs groupes de balises pour identifier les unions et les intersections d'instances à inclure dans un groupe de déploiement. Si vous utilisez un seul groupe de balises, toute instance identifiée par au moins une balise de ce groupe est incluse dans le groupe de déploiement. Si vous utilisez plusieurs groupes de balises, seules sont incluses les instances identifiées par au moins une balise dans chacun des groupes de balises. Pour plus d'informations sur la nouvelle méthode d'ajout d'instances à un groupe de déploiement, consultez Tagging Instances for Deployments . D'autres rubriques ont été mises à jour afin de refléter cette prise en charge, notamment : Création d'une application pour un déploiement sur place (console) , Création d'une application pour un blue/green déploiement (console) , Création d'un groupe de déploiement pour un déploiement sur place (console) , Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site (console) , Deployments et Étape 5 : Création d'une application et d'un groupe de déploiement dans le document Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub .	31 juillet 2017
Rubrique mise à jour	Deux méthodes supplémentaires d'installation de l'CodeDeploy agent sur les instances de Windows Server ont été ajoutées à Installation de l' CodeDeploy agent pour Windows Server . Outre PowerShell les commandes Windows, des instructions sont désormais disponibles pour télécharger le fichier d'installation à l'aide d'un lien HTTPS direct et à l'aide d'une commande de copie Amazon S3. Une fois le fichier téléchargé ou copié sur une instance, vous pouvez exécuter l'installation manuellement.	12 juillet 2017

Modifier	Description	Date de modification
Mise à jour des rubriques	CodeDeploy a amélioré la façon dont il gère les connexions aux GitHub comptes et aux référentiels. Vous pouvez désormais créer et stocker jusqu'à 25 connexions à des GitHub comptes afin d'associer des CodeDeploy applications à des GitHub référentiels. Chaque connexion peut prendre en charge plusieurs référentiels. Vous pouvez créer des connexions à un maximum de 25 GitHub comptes différents, ou créer plusieurs connexions pour un même compte. Une fois que vous avez connecté une application à un GitHub compte, CodeDeploy gère les autorisations d'accès requises sans aucune autre action de votre part. Les mises à jour qui reflètent cette prise en charge ont été apportées à Spécifier les informations relatives à une révision stockée dans un GitHub référentiel , Intégration CodeDeploy avec GitHub et Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub .	30 mai 2017
Mise à jour des rubriques	Dans le passé, si l'agent CodeDeploy détectait des fichiers dans un emplacement cible qui ne faisaient pas partie de la révision de l'application lors du dernier déploiement réussi, le déploiement en cours échouait par défaut. CodeDeploy propose désormais des options concernant la manière dont l'agent gère ces fichiers : échec du déploiement, conservation du contenu ou remplacement du contenu. Créez un déploiement avec CodeDeploy a été mis à jour pour refléter ce support, et la nouvelle section Comportement d'annulation avec du contenu existant a été ajoutée à Redéployez et annulez un déploiement avec CodeDeploy .	16 mai 2017

Modifier	Description	Date de modification
Mise à jour des rubriques	Un Classic Load Balancer dans Elastic Load Balancing peut désormais être attribué à un groupe de déploiement à l'aide de la CodeDeploy console ou. AWS CLI Lors d'un déploiement sur place, un équilibreur de charge empêche que le trafic Internet ne soit acheminé vers une instance alors qu'il est déployé sur cette dernière, puis met de nouveau l'instance à disposition du trafic après la fin du déploiement sur cette dernière. Plusieurs rubriques ont été mises à jour pour refléter cette nouvelle prise en charge, dont Intégration avec d'autres AWS services, Integrating CodeDeploy with Elastic Load Balancing, Création d'une application pour un déploiement sur place (console), Création d'un groupe de déploiement pour un déploiement sur place (console) et AppSpec section « crochets ». Une nouvelle section a été ajoutée au guide de dépannage, Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic.	27 avril 2017

Modifier	Description	Date de modification
Mise à jour des rubriques	Un Classic Load Balancer dans Elastic Load Balancing peut désormais être attribué à un groupe de déploiement à l'aide de la CodeDeploy console ou. AWS CLI Lors d'un déploiement sur place, un équilibreur de charge empêche que le trafic Internet ne soit acheminé vers une instance alors qu'il est déployé sur cette dernière, puis met de nouveau l'instance à disposition du trafic après la fin du déploiement sur cette dernière. Plusieurs rubriques ont été mises à jour pour refléter cette nouvelle prise en charge, dont Intégration avec d'autres AWS services, Integrating CodeDeploy with Elastic Load Balancing, Création d'une application pour un déploiement sur place (console), Création d'un groupe de déploiement pour un déploiement sur place (console) et AppSpec section « crochets ». Une nouvelle section a été ajoutée au guide de dépannage, Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic.	1er mai 2017

Modifier	Description	Date de modification
Mise à jour des rubriques	CodeDeploy est désormais disponible dans la région Chine (Pékin). Pour utiliser les services de la région Chine (Pékin) ou de la région Chine (Ningxia), vous devez disposer d'un compte et d'informations d'identification pour ces régions. Les comptes et informations d'identification AWS des autres régions ne fonctionnent pas pour les régions de Pékin et du Ningxia, et vice versa. Les informations relatives à certaines ressources pour les régions de Chine, telles que les noms des compartiments du kit de CodeDeploy ressources et les procédures d'installation des CodeDeploy agents, ne sont pas incluses dans cette édition du guide de CodeDeploy l'utilisateur. Pour plus d'informations, consultez: • CodeDeploy dans Commencer AWS dans la région de Chine (Pékin) • CodeDeploy Guide de l'utilisateur pour les régions de Chine (version anglaise version chinoise)	29 mars 2017

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	Plusieurs nouvelles rubriques ont été introduites pour refléter la nouvelle CodeDeploy prise en charge blue/green des déploiements, une méthode de déploiement dans laquelle les instances d'un groupe de déploiement (l'environnement d'origine) sont remplacées par un ensemble d'instances différent (l'environnement de remplacement). Vue d'ensemble d'un blue/green déploiement fournit une explication détaillée de la blue/green méthodologie utilisée par CodeDeploy. Les nouvelles rubriques supplémentaires comprennent Création d'une application pour un blue/green déploiement (console), Création d'un groupe de déploiement pour un blue/green déploiement EC2/sur site (console) et Configuration d'un équilibreur de charge dans Elastic Load Balancing pour les déploiements CodeDeploy Amazon EC2. Un grand nombre de rubriques ont également été mises à jour, y compris Créez un déploiement avec CodeDeploy, Utilisation des configurations de déploiement dans CodeDeploy, Créez une application avec CodeDeploy, Utilisation de groupes de déploiement dans CodeDeploy, Travailler avec des déploiements dans CodeDeploy et AppSpec section « crochets ».	25 janvier 2017

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	Une nouvelle rubrique décrit comment authentifier et enregistrer des instances locales à l'aide d'informations d'identification temporaires régulièrement actualisées générées via. Utilisez la register-on-premises-instance commande (ARN de session IAM) pour enregistrer une instance sur site AWS Security Token Service Cette approche prend mieux en charge les importants parcs d'instances sur site que celle utilisant les informations d'identification d'un utilisateur IAM statique sur chaque instance. La rubrique Working with On-Premises Instances a également été mise à jour pour refléter cette nouvelle prise en charge.	28 décembre 2016
Mise à jour des rubriques	CodeDeploy est désormais disponible dans la région Europe (Londres) (eu-west-2). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	13 décembre 2016
Mise à jour des rubriques	CodeDeploy est désormais disponible dans la région du Canada (Centre) (ca-central-1). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	8 décembre 2016
Mise à jour des rubriques	CodeDeploy est désormais disponible dans la région USA Est (Ohio) (us-east-2). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	17 octobre 2016

Modifier	Description	Date de modification
Nouvelles rubriques	Une nouvelle section, Authentification et contrôle d'accès, fournit des informations complètes sur l'utilisation Gestion des identités et des accès AWS (IAM) et CodeDeploy permet de sécuriser l'accès à vos ressources grâce à l'utilisation d'informations d'identification. Ces informations d'identification fournissent les autorisations requises pour accéder aux AWS ressources, telles que la récupération des révisions d'applications depuis les compartiments Amazon S3 et la lecture des balises sur les instances Amazon EC2.	11 octobre 2016
Rubrique mise à jour	Mettre à jour l' CodeDeploy agent sur Windows Server a été mis à jour pour refléter la disponibilité du nouvel CodeDeploy agent de mise à jour pour Windows Server. Lorsqu'il est installé sur une instance Windows Server, le programme de mise à jour vérifie régulièrement la présence de nouvelles versions. Lorsqu'une nouvelle version est détectée, le programme de mise à jour désinstalle la version courante de l'agent, si elle est installée, avant d'installer la version la plus récente.	4 octobre 2016

Modifier	Description	Date de modification
Mise à jour des rubriques	CodeDeploy s'intègre désormais aux CloudWatch alarmes Amazon, ce qui permet d'arrêter un déploiement en cas de modification de l'état d'une alarme spécifiée pendant un certain nombre de périodes consécutives, comme indiqué dans le seuil d'alarme. CodeDeploy prend également désormais en charge l'annulation automatique d'un déploiement si certaines conditions sont remplies, telles qu'un échec du déploiement ou une alarme activée. Un certain nombre de rubriques ont été mises à jour pour refléter ces modifications, notamment Créez une application avec CodeDeploy, Créez un groupe de déploiement avec CodeDeploy, Modifiez les paramètres du groupe de déploiement avec CodeDeploy, Deployments, Redéployez et annulez un déploiement avec CodeDeploy et Intégrations de produits et de services avec CodeDeploy, ainsi qu'une nouvelle rubrique, Surveillance des déploiements avec des CloudWatch alarmes dans CodeDeploy.	le 15 septembre 2016
Rubriques nouvelles et mises à jour	CodeDeploy fournit désormais une intégration avec Amazon CloudWatch Events. Vous pouvez désormais utiliser les CloudWatch événements pour lancer une ou plusieurs actions lorsque des modifications sont détectées dans l'état d'un déploiement ou dans l'état d'une instance appartenant à un groupe de CodeDeploy déployement. Vous pouvez intégrer des actions qui invoquent des AWS Lambda fonctions, qui publient sur des flux Kinesis ou des rubriques Amazon SNS, qui transmettent des messages aux files d'attente Amazon SQS ou qui, à leur tour, déclenchent des actions d'alarme. CloudWatch Pour de plus amples informations, veuillez consulter Surveillance des déploiements avec Amazon CloudWatch Events.	9 septembre 2016

Modifier	Description	Date de modification
Mises à jour des rubriques	Les rubriques Integrating CodeDeploy with Elastic Load Balancing Intégration avec d'autres AWS services ont été mises à jour pour refléter une option d'équilibrage de charge supplémentaire. CodeDeploy prend désormais en charge le Classic Load Balancer et l'Application Load Balancer disponibles dans Elastic Load Balancing.	11 août 2016
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Asie-Pacifique (Mumbai) (ap-south-1). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l'agent CodeDeploy, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région.	27 juin 2016
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Asie-Pacifique (Séoul) (ap-northeast-2). Plusieurs rubriques, y compris celles contenant des instructions pour configurer l'agent CodeDeploy, ont été mises à jour pour refléter la disponibilité de cette nouvelle région. La table des matières a été réorganisée pour inclure des sections pour les instances, les configurations de déploiement, les applications, les groupes de déploiement, les révisions et les déploiements. Une nouvelle section a été ajoutée pour les didacticiels CodeDeploy. Pour une meilleure lisibilité, plusieurs rubriques relativement longues, dont CodeDeploy AppSpec référence de fichier et Résolution des problèmes CodeDeploy, ont été divisées en rubriques plus courtes. Les informations de configuration pour l'agent CodeDeploy ont été placées dans une nouvelle rubrique, CodeDeploy référence de configuration de l'agent.	15 juin 2016

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	La rubrique Codes d'erreur pour AWS CodeDeploy fournit des informations sur certains des messages d'erreur qui peuvent s'afficher en cas d'échec des déploiements CodeDeploy . Les sections suivantes dans la rubrique Résolution des problèmes CodeDeploy ont été mises à jour pour mieux aider à résoudre les problèmes de déploiement : • Les instances EC2 d'un groupe Amazon EC2 Auto Scaling ne se lancent pas et reçoivent l'erreur « Heartbeat Timeout » • Évitez d'associer plusieurs groupes de déploiement à un seul groupe Amazon EC2 Auto Scaling	20 avril 2016
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Amérique du Sud (São Paulo) (sa-east-1). Plusieurs rubriques, y compris celles contenant des instructions pour configurer l'agent CodeDeploy, ont été mises à jour pour refléter la disponibilité de cette nouvelle région. Travailler avec l' CodeDeploy agent a été mise à jour pour refléter la nouvelle option de configuration : max_revisions, que vous utilisez pour spécifier le nombre de révisions d'applications pour un groupe de déploiement que vous souhaitez que l'agent archive. CodeDeploy	10 mars 2016

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	CodeDeploy prend désormais en charge l'ajout de déclencheurs à un groupe de déploiement pour recevoir des notifications concernant les événements liés aux déploiements ou aux instances de ce groupe de déploiement. Ces notifications sont envoyées aux destinataires abonnés à un sujet Amazon Simple Notification Service que vous avez intégré à l'action du déclencheur. Vous pouvez également utiliser les données JSON qui sont créées lorsqu'un déclencheur est activé dans votre propre flux de travail de notification personnalisé. Pour de plus amples informations, veuillez consulter Monitoring Deployments with Amazon SNS Event Notifications. Les procédures ont été mises à jour pour refléter la nouvelle conception de la page Détails de l'application. La section Les déploiements n'échouent pas pendant une heure lorsqu'une instance est mise hors service au cours d'un déploiement de la rubrique Résolution des problèmes CodeDeploy a été mise à jour. CodeDeploy quotas a été mis à jour pour refléter les limites révisées concernant le nombre de groupes de déploiement pouvant être associés à une seule application, les valeurs autorisées pour les paramètres minimaux d'instances saines et les versions requises du AWS SDK pour Ruby.	17 février 2016

Modifier	Description	Date de modification
Rubriques nouvelles et mises à jour	CodeDeploy est désormais disponible dans la région USA Ouest (Californie du Nord) (us-west-1). Plusieurs rubriques , y compris celles contenant des instructions pour configurer l'agent CodeDeploy , ont été mises à jour pour refléter l'ajout de cette nouvelle région. La rubrique Choisissez un type de CodeDeploy référentiel répertorie et décrit les types de référentiel désormais pris en charge par CodeDeploy. Cette nouvelle rubrique sera mise à jour lorsque la prise en charge d'autres types de référentiel sera introduite. Gestion des opérations des CodeDeploy agents a été mis à jour avec des informations sur le nouveau <code>.version</code> fichier ajouté aux instances pour signaler la version actuelle de l' CodeDeploy agent, ainsi que des informations sur les versions prises en charge de l'agent. Une mise en évidence de la syntaxe pour des exemples de code, y compris des exemples JSON et YAML, a été ajoutée au guide de l'utilisateur. Ajouter un fichier de spécification d'application à une révision pour CodeDeploy a été réorganisé sous forme d' step-by-step instructions.	20 janvier 2016

Modifier	Description	Date de modification
Nouvelle rubrique	La rubrique Déployer une application dans un autre AWS compte décrit le processus et les conditions de configuration pour lancer des déploiements qui appartiennent à un autre compte de votre organisation, sans avoir besoin d'un ensemble complet d'informations d'identification pour cet autre compte. Ceci est très utile pour les organisations qui utilisent plusieurs comptes à différentes fins, tels qu'un compte associé aux environnements de développement et de test et un autre associé à l'environnement de production.	30 décembre 2015
Mise à jour de rubrique	La rubrique Intégrations de produits et de services avec CodeDeploy a été refondue. Elle comprend maintenant une section pour des exemples d'intégration issus de la communauté, avec des listes de billets de blog et des exemples de vidéos liés aux intégrations d' CodeDeploy.	le 16 décembre 2015
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Asie-Pacifique (Singapour) (ap-southeast-1). Plusieurs rubriques , y compris celles contenant des instructions pour configurer l'agent CodeDeploy , ont été mises à jour pour refléter la disponibilité de cette nouvelle région.	le 9 décembre 2015
Mises à jour des rubriques	La rubrique Travailler avec l' CodeDeploy agent a été mise à jour pour refléter la nouvelle option : <code>proxy_uri</code> : dans le fichier de configuration de l'agent CodeDeploy. La rubrique CodeDeploy AppSpec référence de fichier a été mise à jour avec des informations sur l'utilisation d'une nouvelle variable d'environnement, <code>DEPLOYMENT_GROUP_ID</code> , à laquelle les scripts de hook peuvent accéder lors d'un événement de cycle de vie de déploiement.	1 décembre 2015

Modifier	Description	Date de modification
Mise à jour de rubrique	Étape 2 : créer un rôle de service pour CodeDeploy a été mis à jour pour refléter une nouvelle procédure de création d'un rôle de service pour CodeDeploy et pour intégrer d'autres améliorations.	13 novembre 2015
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Europe (Francfort) (eu-central-1). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l'agent CodeDeploy, ont été mises à jour pour tenir compte de la disponibilité de cette nouvelle région. La rubrique Résolution des problèmes CodeDeploy a été mise à jour avec des informations sur la façon de s'assurer que les paramètres de temps sont exacts sur les instances.	19 octobre 2015
Nouvelles rubriques	CloudFormation modèles de CodeDeploy référence a été publié pour refléter le nouveau CloudFormation soutien apporté aux CodeDeploy actions. Une rubrique Primary Components a été créée et la définition d'une révision cible a été introduite.	1er octobre 2015
Mises à jour des rubriques	La rubrique Créez un groupe de déploiement avec CodeDeploy a été mise à jour pour refléter la capacité à localiser des instances pour un groupe de déploiement à l'aide de recherches par caractères génériques. La rubrique Instance Health a été mise à jour pour clarifier le concept d'instances saines minimales.	31 août 2015
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Asie-Pacifique (Tokyo) (ap-northeast-1). Plusieurs rubriques, y compris celles contenant des instructions pour configurer l'agent CodeDeploy, ont été mises à jour pour refléter la disponibilité de cette nouvelle région.	19 août 2015

Modifier	Description	Date de modification
Mises à jour des rubriques	CodeDeploy prend désormais en charge les déploiements sur les instances locales de Red Hat Enterprise Linux (RHEL) et les instances Amazon EC2. Pour plus d'informations, consultez les rubriques suivantes : • Systèmes d'exploitation pris en charge par l' agent CodeDeploy • Utilisation d'instances pour CodeDeploy • Tutoriel : Déploiement WordPress sur une instance Amazon EC2 (Amazon Linux ou Red Hat Enterprise Linux et Linux, macOS ou Unix) • Tutoriel : Déployer une application sur une instance locale avec CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux)	23 juin 2015
Mise à jour de rubrique	CodeDeploy fournit désormais un ensemble de variables d'environnement que vos scripts de déploiement peuvent utiliser lors des déploiements. Ces variables d'environnement incluent des informations telles que le nom de l' application CodeDeploy actuelle, le groupe de déploiement, l'événement du cycle de vie du déploiement, ainsi que l'identifiant de CodeDeploy déploiement actuel. Pour plus d'informations, consultez la fin de la section AppSpec section « crochets » dans la rubrique CodeDeploy AppSpec référence de fichier.	29 mai 2015

Modifier	Description	Date de modification
Mises à jour des rubriques	CodeDeploy fournit désormais un ensemble de politiques AWS gérées dans IAM que vous pouvez utiliser au lieu de créer vous-même les politiques équivalentes manuellement. Il s'agit des licences suivantes : • Une politique permettant à un utilisateur d'enregistrer des révisions CodeDeploy uniquement auprès de celui-ci, puis de les déployer via CodeDeploy. • Politique visant à fournir à un utilisateur un accès complet aux CodeDeploy ressources. • Politique visant à fournir à un utilisateur un accès en lecture seule aux CodeDeploy ressources. • Une politique à associer à un rôle de service afin d'identifier les instances Amazon EC2 à l'aide de leurs balises Amazon EC2, de leurs balises d'instance sur site ou des noms de groupes Amazon EC2 Auto Scaling et de déployer des révisions d'application en conséquence. CodeDeploy Pour plus d'informations, consultez la section Exemples de politiques gérées par le client Authentification et contrôle d'accès.	29 mai 2015
Mises à jour des rubriques	CodeDeploy est désormais disponible dans la région Europe (Irlande) (eu-west-1) et dans la région Asie-Pacifique (Sydney) (ap-southeast-2). Plusieurs rubriques, notamment celles contenant les instructions de configuration de l' CodeDeploy agent, ont été mises à jour pour tenir compte de la disponibilité de ces nouvelles régions.	7 mai 2015

Modifier	Description	Date de modification
Nouvelles rubriques	CodeDeploy prend désormais en charge les déploiements sur des instances sur site et des instances Amazon EC2. Les rubriques suivantes ont été ajoutées pour décrire cette nouvelle prise en charge : • Working with On-Premises Instances • Tutoriel : Déployer une application sur une instance locale avec CodeDeploy (Windows Server, Ubuntu Server ou Red Hat Enterprise Linux) • Working with On-Premises Instances	2 avril 2015
Nouvelle rubrique	La rubrique CodeDeploy ressources a été ajoutée.	2 avril 2015
Mise à jour de rubrique	La rubrique Résolution des problèmes CodeDeploy a été mise à jour : • Une nouvelle section Des processus de longue durée peuvent entraîner l'échec des déploiements décrit la procédure que vous pouvez entreprendre pour identifier et résoudre les échecs de déploiement causés par des processus longs. • La Résolution générale des problèmes liés à Amazon EC2 Auto Scaling section a été mise à jour pour indiquer que la logique de temporisation Amazon EC2 Auto Scaling pour CodeDeploy l'agent CodeDeploy a été augmentée de cinq minutes à une heure. • Une nouvelle Des hooks de cycle de vie Amazon EC2 Auto Scaling mal adaptés peuvent entraîner l'arrêt ou l'échec des déploiements automatiques vers les groupes Amazon EC2 Auto Scaling section décrit les étapes à suivre pour identifier et traiter les échecs de déploiements automatiques vers les groupes Amazon EC2 Auto Scaling.	2 avril 2015

Modifier	Description	Date de modification
Mises à jour des rubriques	Les rubriques suivantes ont été mises à jour pour refléter les nouvelles recommandations relatives à la création de vos propres politiques personnalisées, puis à leur association aux utilisateurs et aux rôles dans IAM : • Configurer une instance Amazon EC2 pour qu'elle fonctionne avec CodeDeploy • Étape 4 : Création d'un profil d'instance IAM pour vos instances Amazon EC2 • Étape 2 : créer un rôle de service pour CodeDeploy Deux sections ont été ajoutées à la rubrique Résolution des problèmes CodeDeploy : • Liste de contrôle de dépannage général • PowerShell Les scripts Windows n'utilisent pas la version 64 bits de Windows PowerShell par défaut La section AppSpec section « crochets » dans la rubrique CodeDeploy AppSpec référence de fichier a été mise à jour pour décrire plus précisément les événements de cycle de vie de déploiement disponibles.	12 février 2015
Mises à jour des rubriques	Une nouvelle section a été ajoutée à la rubrique Résolution des problèmes CodeDeploy : Les instances EC2 d'un groupe Amazon EC2 Auto Scaling ne se lancent pas et reçoivent l'erreur « Heartbeat Timeout ». Une CloudBees section a été ajoutée à Intégrations de produits et de services avec CodeDeploy.	28 janvier 2015

Modifier	Description	Date de modification
Mises à jour des rubriques	Les sections suivantes ont été ajoutées à la rubrique Résolution des problèmes CodeDeploy : • L'utilisation de certains éditeurs de texte pour créer des AppSpec fichiers et des scripts shell peut entraîner l'échec des déploiements. • L'utilisation du Finder sur MacOS pour grouper une révision d'application peut entraîner l'échec des déploiements • Résolution des problèmes liés à un échec ApplicationStop ou à un événement lié AfterBlockTraffic au cycle de vie du déploiement BeforeBlockTraffic • Résolution des problèmes liés à un échec du cycle de vie d'un DownloadBundle déploiement avec UnknownError : non ouvert pour lecture • Résolution générale des problèmes liés à Amazon EC2 Auto Scaling	20 janvier 2015
Nouvelles rubriques	La section Intégrations de produits et de services avec CodeDeploy a été mise à jour pour inclure les rubriques suivantes : • Intégration CodeDeploy à Amazon EC2 Auto Scaling • Tutoriel : CodeDeploy À utiliser pour déployer une application dans un groupe Auto Scaling • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Intégration CodeDeploy avec GitHub • Tutoriel : CodeDeploy À utiliser pour déployer une application à partir de GitHub	9 janvier 2015

Modifier	Description	Date de modification
Mises à jour des rubriques	- La section Déployez automatiquement à partir CodePipeline de CodeDeploy a été ajoutée à la rubrique Intégration CodeDeploy avec GitHub. Vous pouvez désormais déclencher automatiquement un déploiement à partir d'un GitHub référentiel chaque fois que le code source de ce référentiel est modifié. - La section Résoudre les problèmes liés à Amazon EC2 Auto Scaling a été ajoutée à la rubrique Résolution des problèmes CodeDeploy. Cette nouvelle section explique comment résoudre les problèmes courants liés au déploiement sur des groupes Amazon EC2 Auto Scaling. - La nouvelle sous-section « Exemples de fichiers » a été ajoutée à la section AppSpec section « fichiers » (déploiements EC2/sur site uniquement) de la rubrique CodeDeploy AppSpec référence de fichier. Cette nouvelle sous-section inclut plusieurs exemples d'utilisation de la <code>files</code> section d'un AppSpec fichier pour indiquer CodeDeploy de copier des fichiers ou des dossiers spécifiques vers des emplacements spécifiques sur une instance Amazon EC2 lors d'un déploiement.	8 janvier 2015
Nouvelle rubrique	Monitoring Deployments a été ajouté. CodeDeploy est intégré à AWS CloudTrail un service qui capture les appels d'API effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux CodeDeploy dans un compartiment Amazon S3 que vous spécifiez.	17 décembre 2014
Première version publique	Il s'agit de la première version publique du Guide de l'utilisateur CodeDeploy .	12 novembre 2014

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le [AWS glossaire](#) dans la Glossaire AWS référence.

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.