

Guida per l'utente

AWS CodeBuild



Versione API 2016-10-06

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

AWS CodeBuild: Guida per l'utente

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è AWS CodeBuild?	1
Come eseguire CodeBuild	1
Prezzi per CodeBuild	3
Come posso iniziare CodeBuild?	3
Concetti	3
Come CodeBuild funziona	3
Fasi successive	5
Nozioni di base	6
Nozioni di base utilizzando la console	6
Fase 1: Creare il codice sorgente	7
Passaggio 2: creare il file buildspec	10
Fase 3: Creare due bucket S3	12
Fase 4: caricamento del codice di origine e del file buildspec	13
Fase 5: creazione del progetto di compilazione	14
Fase 6: esecuzione della compilazione	16
Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione	17
Fase 8: visualizzazione di informazioni dettagliate sulla compilazione	18
Fase 9: generazione dell'artefatto di output della compilazione	19
Passaggio 10: eliminare i bucket S3	20
Wrapping	20
Nozioni di base per l'utilizzo della AWS CLI	21
Fase 1: Creare il codice sorgente	22
Passaggio 2: creare il file buildspec	25
Fase 3: Creare due bucket S3	27
Fase 4: caricamento del codice di origine e del file buildspec	28
Fase 5: creazione del progetto di compilazione	29
Fase 6: esecuzione della compilazione	33
Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione	35
Fase 8: visualizzazione di informazioni dettagliate sulla compilazione	38
Fase 9: generazione dell'artefatto di output della compilazione	40
Passaggio 10: eliminare i bucket S3	41
Wrapping	42
Esempi basati sui casi d'uso	43
Esempi interservizi	44

Esempio di Amazon ECR	45
Esempio di Amazon EFS	52
AWS CodePipeline campioni	58
AWS Config campione	69
Esempio di notifiche di compilazione	71
Creazione di esempi di badge	86
Crea un progetto di compilazione con i badge di costruzione	87
Accedi ai badge di costruzione AWS CodeBuild	90
Pubblica badge di costruzione CodeBuild	91
CodeBuild stati dei badge	91
Esempio di rapporto di prova	92
Esegui l'esempio del rapporto di prova	92
Esempi Docker per CodeBuild	99
Docker in un esempio di immagine personalizzata	99
Esempio di server di creazione di immagini Docker	102
Esempio di build di Windows Docker	105
Esempio «Pubblica immagine Docker su Amazon ECR'	107
Registro privato con Gestione dei segreti AWS campione	116
Output di compilazione ospitato in un bucket S3	120
Esempio di ingressi e uscite multipli	124
Crea un progetto di compilazione con più input e output	125
Crea un progetto senza una fonte	128
Versioni runtime nel file builds spec di esempio	128
Aggiorna la versione di runtime nel file builds spec	129
Specifica di due runtime	133
Esempio di versione di origine	137
Specificare una versione GitHub del repository con un ID di commit	139
Specificare una versione GitHub del repository con un riferimento e un ID di commit	141
Esempi di repository di sorgenti di terze parti	141
Esegui l'esempio di Bitbucket	142
Esegui l'esempio di GitHub Enterprise Server	148
Esegui l' GitHub esempio di filtro pull request e webhook	154
Tutorial: firma del codice Apple con Fastlane sull' CodeBuild utilizzo di S3 per l'archiviazione dei certificati	159
Tutorial: firma del codice Apple con Fastlane in CodeBuild Utilizzo GitHub per l'archiviazione dei certificati	165

Imposta i nomi degli artefatti in fase di compilazione	171
Esegui esempi di Windows	175
Esegui gli esempi di Windows	175
Struttura delle directory	176
F# e .NET Framework	176
Visual Basic e .NET Framework	177
File	177
F# e .NET Framework	177
Visual Basic e .NET Framework	182
Pianifica una compilazione	196
Riferimento per buildspec	199
Nome del file buildspec e posizione di storage	199
Sintassi buildspec	200
version	203
run-as	203
env	204
proxy	209
phases	209
reports	213
artefatti	216
cache	222
Esempio di buildspec	224
Versioni di buildspec	227
Riferimento alle specifiche di compilazione in Batch	228
batch	228
batch/build-graph	228
batch/build-list	231
batch/build-matrix	234
batch/build-fanout	236
Riferimento ambiente di compilazione	238
Immagini Docker fornite da CodeBuild	239
Ottieni l'elenco delle immagini Docker correnti	239
EC2 calcola immagini	240
Immagini di calcolo Lambda	241
Immagini obsolete CodeBuild	246
Runtime disponibili	247

Versioni di runtime	266
Modi e tipi di calcolo dell'ambiente di creazione	271
Informazioni sull'elaborazione	271
Informazioni sui tipi di ambienti con capacità riservata	272
Informazioni sui tipi di ambiente on-demand	325
Shell e comandi negli ambienti di compilazione	337
Variabili di ambiente degli ambienti di compilazione	338
Attività in background degli ambienti di compilazione	344
Costruisci progetti	346
Creazione di un progetto di compilazione	346
Prerequisiti	347
Creare un progetto di compilazione (console)	347
Creazione di un progetto di compilazione (AWS CLI)	369
Creazione di un progetto di compilazione (AWS SDKs)	390
Creazione di un progetto di compilazione (CloudFormation)	390
Creazione di una regola di notifica	390
Modifica le impostazioni del progetto di compilazione	394
Modifica delle impostazioni di un progetto di compilazione (console)	394
Modifica delle impostazioni di un progetto di compilazione (AWS CLI)	418
Modifica delle impostazioni di un progetto di compilazione (AWS SDKs)	419
Token di accesso multiplo	420
Passaggio 1: Creare un segreto o una CodeConnections connessione di Secrets Manager	420
Fase 2: concedere al ruolo IAM CodeBuild del progetto l'accesso ai segreti di Secrets Manager	421
Fase 3: Configurare Secrets Manager o i CodeConnections token	423
Opzioni di configurazione aggiuntive	427
Eliminare progetti di compilazione	430
Eliminazione di un progetto di compilazione (console)	431
Eliminazione di un progetto di compilazione (AWS CLI)	431
Eliminazione di un progetto di compilazione (AWS SDKs)	432
Ottieni un progetto di costruzione pubblico URLs	432
Condividi progetti di costruzione	433
Condividere un progetto	434
Servizi correlati	437
Accedi ai progetti condivisi	437

Annullare la condivisione di un progetto condiviso	437
Identifica un progetto condiviso	438
Autorizzazioni per i progetti condivisi	438
Etichetta i progetti di costruzione	438
Aggiunta di un tag a un progetto	440
Visualizzazione dei tag di un progetto	441
Modifica dei tag di un progetto	442
Rimozione di un tag da un progetto	443
Usa i corridori	444
GitHub Azioni	445
GitLab corridori	467
Buildkite runner	482
Usa i webhook	504
Procedure consigliate per l'utilizzo dei webhook	505
Eventi webhook Bitbucket	506
GitHub webhook globali e organizzativi	519
GitHub webhook manuali	526
GitHub eventi webhook	527
GitLab webhook di gruppo	544
GitLab webhook manuali	549
GitLab eventi webhook	550
Webhook manuali Buildkite	565
Pull richiedi l'approvazione dei commenti	566
Visualizza i dettagli del progetto di costruzione	572
Visualizzazione dei dettagli di un progetto di compilazione (console)	572
Visualizzazione dei dettagli di un progetto di compilazione (AWS CLI)	573
Visualizzazione dei dettagli di un progetto di compilazione (AWS SDKs)	575
Visualizza i nomi dei progetti di compilazione	575
Visualizzazione di un elenco di nomi di progetti di compilazione (console)	576
Visualizzazione di un elenco di nomi di progetti di compilazione (AWS CLI)	576
Visualizzazione di un elenco di nomi di progetti di compilazione (AWS SDKs)	578
Compilazioni	579
Esegui le build manualmente	580
Esegui una build localmente	580
Esecuzione di una compilazione (console)	584
Esecuzione di una compilazione (AWS CLI)	585

Esegui una build in batch (AWS CLI)	592
Avvio di esecuzioni di compilazione automaticamente (AWS CLI)	594
Interruzione di esecuzioni di compilazioni automaticamente (AWS CLI)	595
Esecuzione di una compilazione (AWS SDKs)	595
Esegui build su elaborazione Lambda	596
Quali strumenti e runtime saranno inclusi nelle immagini docker dell'ambiente di runtime curato su cui vengono eseguiti? AWS Lambda	596
Cosa succede se l'immagine curata non include gli strumenti di cui ho bisogno?	596
In quali aree geografiche è AWS Lambda supportata l'elaborazione? CodeBuild	597
Limitazioni di AWS Lambda calcolo	598
Implementa una funzione Lambda utilizzando AWS SAM con Lambda Java CodeBuild	598
Crea un'app React a pagina singola con CodeBuild Lambda Node.js	602
Aggiornare la configurazione di una funzione Lambda con CodeBuild Lambda Python	605
Run si basa su flotte a capacità riservata	609
Crea un parco veicoli a capacità riservata	611
Best practice	613
Posso condividere una flotta di capacità riservata tra più progetti? CodeBuild	613
Come funziona il calcolo basato sugli attributi?	613
Posso specificare manualmente un'istanza Amazon EC2 per la mia flotta?	614
Quali regioni supportano flotte a capacità riservata?	614
Come posso configurare una flotta macOS a capacità riservata?	615
Come posso configurare un'Amazon Machine Image (AMI) personalizzata per un parco macchine a capacità riservata?	616
Limitazioni delle flotte a capacità riservata	617
Proprietà del parco veicoli a capacità riservata	618
Esempi di capacità riservata	622
Esegui build in batch	624
Ruolo di sicurezza	625
Tipi di build in batch	625
Modalità report Batch	629
Ulteriori informazioni	629
Esegui test paralleli	630
Support in AWS CodeBuild	631
Abilita l'esecuzione di test paralleli nelle build in batch	634
Usa il comando <code>codebuild-tests-run</code> CLI	635
Usa il comando <code>codebuild-glob-search</code> CLI	638

Informazioni sulla suddivisione dei test	639
Unisci automaticamente i report di build individuali	640
Esempi di esecuzione di test paralleli	643
Compilazioni di cache	654
Memorizzazione nella cache di Amazon S3	654
Caching locale	661
Specificare una cache locale	662
Compilazioni di debug	665
Esegui il debug delle build con sandbox CodeBuild	665
Compilazioni di debug con Session Manager	666
Compilazioni di debug con sandbox CodeBuild	666
Compilazioni di debug con Session Manager	696
Eliminazione delle compilazioni	701
Eliminazione delle compilazioni (AWS CLI)	701
Eliminazione delle compilazioni (AWS SDKs)	702
Riprova le build manualmente	703
Riprova una compilazione manualmente (console)	703
Riprova una build manualmente (AWS CLI)	703
Riprova una build manualmente (AWS SDKs)	704
Riprova le build automaticamente	704
Riprova automaticamente una build (console)	705
Riprova una build automaticamente (AWS CLI)	705
Riprova automaticamente una build (AWS SDKs)	706
Stop alle costruzioni	706
Interruzione di una compilazione (console)	706
Interruzione di una compilazione (AWS CLI)	707
Interruzione di una compilazione (AWS SDKs)	707
Interrompi le compilazioni in batch	708
Interrompi una compilazione in batch (console)	708
Interrompi una compilazione in batch (AWS CLI)	709
Interrompi una compilazione in batch (AWS SDKs)	709
Trigger si crea automaticamente	709
Creazione di trigger della compilazione	709
Modifica dei trigger di compilazione	713
Visualizzazione dei dettagli delle compilazioni	716
Visualizzazione dei dettagli delle compilazioni (console)	716

Visualizzazione dei dettagli delle compilazioni (AWS CLI)	717
Visualizzazione dei dettagli delle compilazioni (AWS SDKs)	718
Transizioni delle fasi della compilazione	718
Visualizza build IDs	718
Visualizza un elenco di build IDs (console)	718
Visualizza un elenco di build () IDs AWS CLI	719
Visualizza un elenco di build in batch IDs (AWS CLI)	720
Visualizza un elenco di build IDs (AWS SDKs)	722
Visualizza la build IDs di un progetto di compilazione	722
Visualizza un elenco di build IDs per un progetto di compilazione (console)	722
Visualizza un elenco di build IDs per un progetto di compilazione (AWS CLI)	722
Visualizza un elenco di build in batch IDs per un progetto di compilazione (AWS CLI)	724
Visualizza un elenco di build IDs per un progetto di compilazione (AWS SDKs)	725
Rapporti sui test	726
Crea report di test	727
Crea report sulla copertura del codice	728
Crea un rapporto sulla copertura del codice	729
Scopri automaticamente i report	730
Configura l'individuazione automatica dei report utilizzando la console	732
Configura l'individuazione automatica dei report utilizzando le variabili di ambiente del progetto	732
Gruppi di report	733
Creazione di un gruppo di report	734
Denominazione dei gruppi di report	739
Condividi gruppi di report	740
Specifica dei file di test	746
Specifica dei comandi di test	747
Applicazione di un tag a un gruppo di report	747
Aggiornamento di un gruppo di report	753
Framework di test	757
Configura Jasmine	757
Configura Jest	759
Configura pytest	761
Configurare RSpec	762
Visualizzazione dei report di test	762
Visualizzazione dei report di test in una compilazione	763

Visualizzazione dei report di test in un gruppo di report	764
Visualizza i report dei test nel tuo account AWS	764
Autorizzazioni per il rapporto di prova	764
Ruolo IAM per i report di test	764
Autorizzazioni per le operazioni correlate ai report di test	766
Esempi di autorizzazioni per i report di test	767
Stati del rapporto di test	767
Supporto per VPC	769
Casi d'uso	769
Le migliori pratiche per VPCs	770
Limitazioni di VPCs	771
Consenti l'accesso ad Amazon VPC nei tuoi progetti CodeBuild	771
Risolvi i problemi relativi alla configurazione del VPC	772
Utilizzo degli endpoint VPC	773
Prima di creare endpoint VPC	773
Crea endpoint VPC per CodeBuild	774
Crea una policy per gli endpoint VPC per CodeBuild	774
Utilizza un server proxy CodeBuild gestito	775
Configurare una configurazione proxy gestita per flotte a capacità riservata	776
Gestisci un parco veicoli con capacità CodeBuild riservata	777
Utilizza un server proxy	777
Configura i componenti necessari per l'esecuzione CodeBuild in un server proxy	778
Esegui CodeBuild in un server proxy esplicito	781
Esegui CodeBuild in un server proxy trasparente	785
Esegui un programma di gestione dei pacchetti e di altri strumenti in un server proxy	787
CloudFormation Modello VPC	789
Registrazione di log e monitoraggio	796
Registra le chiamate CodeBuild API	796
AWS CodeBuild Informazioni in CloudTrail	796
Informazioni sulle voci dei file di AWS CodeBuild registro	797
Monitora le build	800
CloudWatch metriche	801
CloudWatch metriche di utilizzo delle risorse	803
CloudWatch dimensioni	805
CloudWatch allarmi	805
Visualizza le metriche CodeBuild	806

Visualizza le metriche sull' CodeBuild utilizzo delle risorse	808
Crea allarmi CodeBuild in CloudWatch	812
Sicurezza	814
Protezione dei dati	815
Crittografia dei dati	816
Gestione delle chiavi	817
Privacy del traffico	818
Gestione dell'identità e degli accessi	818
Panoramica sulla gestione degli accessi	818
Utilizzo di policy basate su identità	823
AWS CodeBuild riferimento alle autorizzazioni	852
Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse	859
Visualizzazione di risorse nella console	863
Convalida della conformità	864
Resilienza	864
Sicurezza dell'infrastruttura	864
Accesso al provider di origine	865
Crea e archivia un token in un segreto di Secrets Manager	865
GitHub e accesso a GitHub Enterprise Server	868
Accesso a Bitbucket	880
GitLab accesso	889
Prevenzione del confused deputy tra servizi	896
Argomenti avanzati	899
Consenti agli utenti di interagire con CodeBuild	899
Consenti CodeBuild di interagire con altri servizi AWS	907
Crittografa gli output di compilazione	915
Interagisci con l' CodeBuild utilizzo di AWS CLI	917
Guida di riferimento alla riga di comando	918
AWS SDKs e riferimento agli strumenti	919
Supportato AWS SDKs e strumenti per AWS CodeBuild	919
Lavorare con AWS SDKs	920
Specificare l'endpoint CodeBuild	921
Specificate l' AWS CodeBuild endpoint (AWS CLI)	922
Specificare l' AWS CodeBuild endpoint (AWS SDK)	922
Usa CodeBuild con CodePipeline	925
Prerequisiti	926

Creazione di una pipeline (console)	928
Crea una pipeline ()AWS CLI	932
Aggiungi un'azione di compilazione	937
Aggiungi un'azione di test	941
Utilizzare CodeBuild con Codecov	945
Integrazione di Codecov in un progetto di compilazione	945
Da usare CodeBuild con Jenkins	948
Configura Jenkins	949
Installazione del plug-in	949
Usa il plugin	949
Utilizzalo CodeBuild con app serverless	952
Risorse correlate	952
Avvisi di terze parti	952
1) immagine Docker di base: windowsservercore	953
2) immagine Docker basata su Windows-choco	954
3) immagine Docker basata su Windows-git --version 2.16.2	955
4) immagine Docker basata su Windows— --versione 15.0.26320.2 microsoft-build-tools	955
5) Immagine Docker basata su Windows: nuget.commandline - versione 4.5.1	959
7) immagine Docker basata su Windows: netfx-4.6.2-devpack	960
8) immagine Docker basata su Windows: visualfsharpools, v 4.0	961
9) immagine Docker basata su Windows— -4.6 netfx-pcl-reference-assemblies	962
10) Immagine Docker basata su Windows: visualcppbuildtools v 14.0.25420.1	966
11) Immagine Docker basata su Windows: 3-ondemand-package.cab microsoft-windows-netfx	970
12) immagine Docker basata su Windows: dotnet-sdk	971
Usa le chiavi di CodeBuild condizione come variabili del ruolo del servizio IAM	972
AWS CodeBuild tasti di condizione	973
Applica le impostazioni di connettività VPC ai tuoi progetti e alle tue flotte	973
Impedisci modifiche non autorizzate al progetto buildspect	974
Limita i tipi di calcolo per le tue build	975
Controlla le impostazioni delle variabili d'ambiente	975
Usa le variabili nei nomi delle chiavi delle condizioni	977
Verifica l'esistenza di attributi nelle richieste API	977
Esempi di codice	979
Nozioni di base	979
Azioni	980

Risoluzione dei problemi	997
Apache Maven compila artefatti di riferimento dal repository errato	998
I comandi di compilazione vengono eseguiti come root per impostazione predefinita	1000
Le compilazioni potrebbero fallire quando i nomi dei file non sono U.S. Caratteri inglesi	1000
Le build potrebbero non riuscire quando si ottengono i parametri da Amazon EC2 Parameter Store	1001
Impossibile accedere al filtro branch nella console CodeBuild	1002
Impossibile visualizzare l'esito positivo o negativo della compilazione	1003
Lo stato della build non viene segnalato al fornitore di origine	1003
Impossibile trovare e selezionare l'immagine di base della piattaforma Windows Server Core 2019	1003
I comandi precedenti nei file buildspec non vengono riconosciuti dai comandi successivi	1004
Errore "Access denied" (Accesso negato) durante il tentativo di eseguire il download della cache	1004
Errore "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" durante l'utilizzo di un'immagine di compilazione personalizzata	1005
Errore: «Il contenitore di compilazione è stato trovato morto prima di completare la compilazione. Il contenitore di compilazione è morto perché aveva esaurito la memoria o l'immagine Docker non è supportata. ErrorCode: 500"	1006
Errore: "Impossibile connettersi al daemon Docker" quando si esegue una build	1006
Errore: "non CodeBuild è autorizzato a eseguire: sts:AssumeRole" durante la creazione o l'aggiornamento di un progetto di compilazione	1008
Errore: «Errore nella chiamata GetBucketAcl: il proprietario del bucket è cambiato o il ruolo del servizio non è più autorizzato a chiamare s3:» GetBucketAcl	1009
Errore "Failed to upload artifacts: Invalid arn" (Impossibile caricare artefatti: ARN non valido) durante l'esecuzione di una compilazione	1009
Errore "Git Clone Failed: unable to access 'your-repository-URL': SSL certificate problem: self signed certificate" (Git Clone non riuscito: impossibile accedere a "your-repository-URL": problema di certificato SSL: certificato autofirmato)	1009
Errore "The bucket you are attempting to access must be addressed using the specified endpoint" (Il bucket a cui tenti di accedere deve essere individuato tramite l'endpoint specificato) durante l'esecuzione di una compilazione	1010
Errore: "This build image requires selecting at least one runtime version." (Questa immagine di compilazione richiede la selezione di almeno una versione runtime.)	1011
Errore: "QUEUED: INSUFFICIENT_SUBNET" quando una compilazione ha esito negativo in una coda di compilazione	1012

Errore: «Impossibile scaricare la cache: RequestError: Invio della richiesta non riuscito a causa di: x509: impossibile caricare le radici del sistema e nessuna root fornita»	1012
Errore: «Impossibile scaricare il certificato da S3. AccessDenied»	1013
Errore "Unable to Locate Credentials" (Impossibile individuare le credenziali)	1013
RequestError errore di timeout durante l'esecuzione CodeBuild in un server proxy	1015
La bourne shell (sh) deve esistere nelle immagini di compilazione	1016
Avvertenza: "Skipping install of runtimes. Runtime version selection is not supported by this build image" (L'installazione dei runtime non viene eseguita. L'immagine di compilazione non supporta la selezione delle versioni dei runtime) durante l'esecuzione di una compilazione	1017
Errore: «Impossibile verificare l' JobWorker identità»	1017
Avvio della compilazione non riuscito	1017
Accesso ai GitHub metadati nelle build memorizzate nella cache locale	1018
AccessDenied: Il proprietario del bucket per il gruppo di report non corrisponde al proprietario del bucket S3... ..	1018
Errore: «Le tue credenziali non dispongono di uno o più ambiti di privilegi richiesti» durante la creazione di un progetto con CodeBuild CodeConnections	1019
Errore: «Siamo spiacenti, non è stato richiesto alcun terminale, impossibile ricevere input» durante la compilazione con il comando di installazione di Ubuntu	1020
Quote	1022
Quote di servizio	1022
Altri limiti	1028
Costruisci progetti	1028
Costruisce	1028
Flotte di calcolo	1029
Report	1030
Tag	1031
Cronologia dei documenti	1032
Aggiornamenti precedenti	1056
.....	mlxx

Che cos'è AWS CodeBuild?

AWS CodeBuild è un servizio di compilazione completamente gestito nel cloud. CodeBuild compila il codice sorgente, esegue test unitari e produce artefatti pronti per l'implementazione. CodeBuild elimina la necessità di fornire, gestire e scalare i propri server di build. Fornisce ambienti di compilazione predefiniti per i linguaggi di programmazione più diffusi e strumenti di compilazione come Apache Maven, Gradle e molti altri. Puoi anche personalizzare gli ambienti di compilazione CodeBuild per utilizzare i tuoi strumenti di compilazione. CodeBuild si ridimensiona automaticamente per soddisfare i picchi di richieste di compilazione.

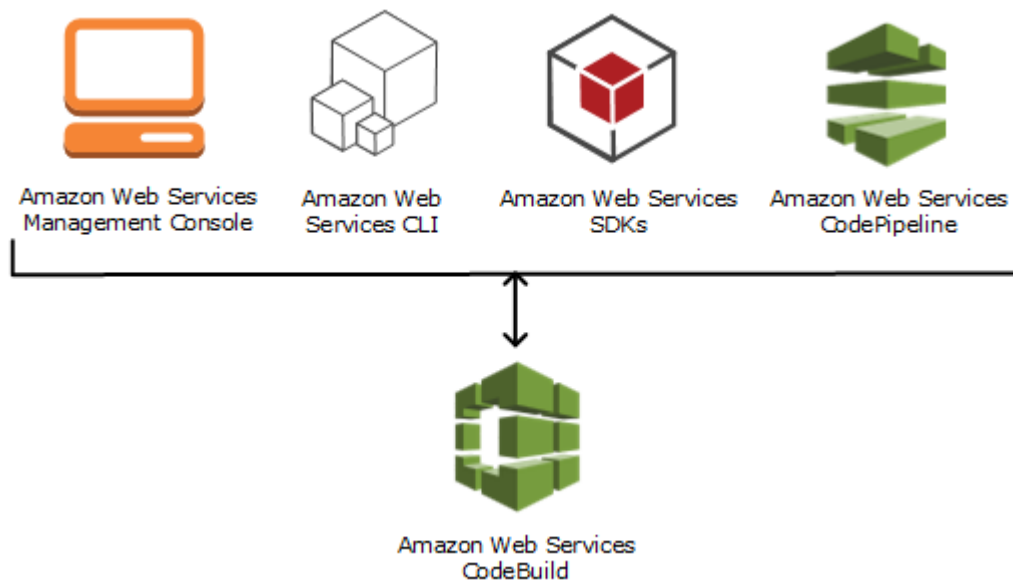
CodeBuild offre i seguenti vantaggi:

- Completamente gestito: CodeBuild elimina la necessità di configurare, applicare patch, aggiornare e gestire i propri server di build.
- Su richiesta: CodeBuild scalabilità su richiesta per soddisfare le esigenze di costruzione. Paghi soltanto per il tempo di compilazione utilizzato.
- Pronto all'uso: CodeBuild fornisce ambienti di compilazione preconfigurati per i linguaggi di programmazione più diffusi. Devi semplicemente puntare allo script di compilazione per avviare la tua prima build.

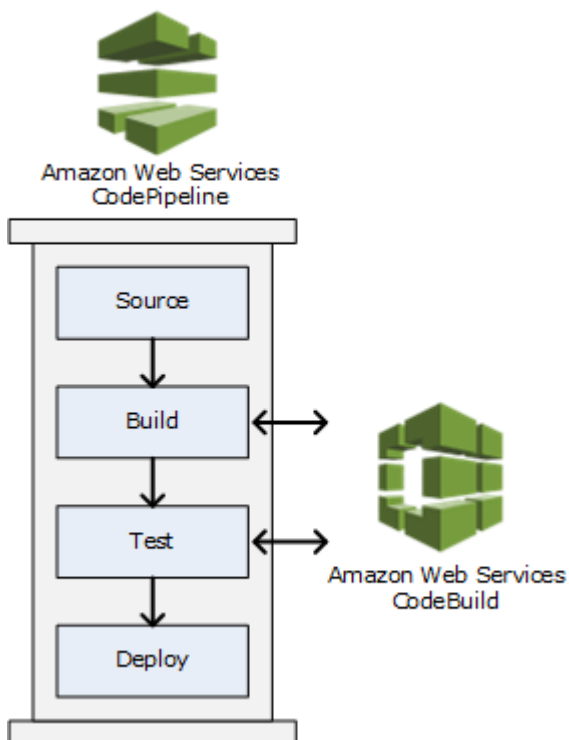
Per ulteriori informazioni, consulta [AWS CodeBuild](#).

Come eseguire CodeBuild

Puoi usare la AWS CodePipeline console AWS CodeBuild o per eseguire CodeBuild. Puoi anche automatizzare l'esecuzione di CodeBuild utilizzando AWS Command Line Interface (AWS CLI) o. AWS SDKs



Come mostra il diagramma seguente, è possibile aggiungere CodeBuild come azione di compilazione o test alla fase di compilazione o test di una pipeline in. AWS CodePipeline AWS CodePipeline è un servizio di distribuzione continua che puoi utilizzare per modellare, visualizzare e automatizzare i passaggi necessari per rilasciare il codice. Tra le funzionalità è incluso lo sviluppo del codice. Una pipeline è una struttura di flusso di lavoro che descrive in che modo le modifiche di un codice passano attraverso il processo di rilascio.



Per CodePipeline utilizzarlo per creare una pipeline e quindi aggiungere un'azione di CodeBuild compilazione o test, vedi. [Usa CodeBuild con CodePipeline](#) Per ulteriori informazioni in merito CodePipeline, consulta la [Guida per l'AWS CodePipeline utente](#).

La CodeBuild console offre anche un modo per cercare rapidamente le risorse, come repository, progetti di creazione, applicazioni di distribuzione e pipeline. Scegli Go to resource (Vai alla risorsa) o premi il tasto / e immetti il nome della risorsa. Qualsiasi corrispondenza verrà visualizzata nell'elenco. Le ricerche rispettano la distinzione tra maiuscole e minuscole. Puoi visualizzare solo le risorse per le quali disponi dell'autorizzazione di visualizzazione. Per ulteriori informazioni, consulta [Visualizzazione di risorse nella console](#).

Prezzi per CodeBuild

Per informazioni, consulta [Prezzi di CodeBuild](#).

Come posso iniziare CodeBuild?

È consigliabile completare la procedura seguente:

1. Scopri di più CodeBuild leggendo le informazioni in [Concetti](#).
2. Sperimenta CodeBuild in uno scenario di esempio seguendo le istruzioni riportate in [Nozioni di base utilizzando la console](#).
3. CodeBuild Utilizzalo nei tuoi scenari seguendo le istruzioni riportate in [Pianifica una compilazione](#).

AWS CodeBuild concetti

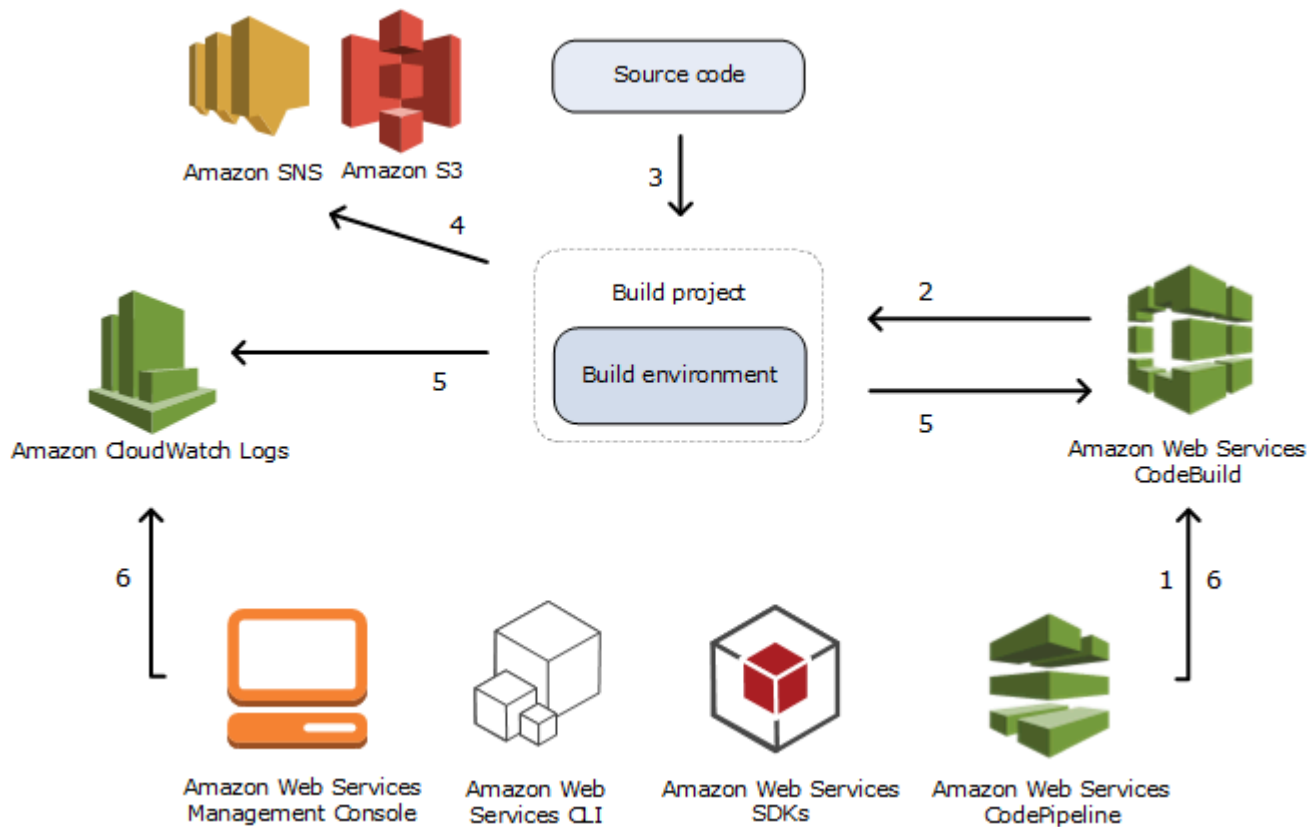
I seguenti concetti sono importanti per capire come CodeBuild funziona.

Argomenti

- [Come CodeBuild funziona](#)
- [Fasi successive](#)

Come CodeBuild funziona

Il diagramma seguente mostra cosa succede quando si esegue una build con CodeBuild:



1. Come input, devi fornire un progetto CodeBuild di compilazione. Un progetto di compilazione include informazioni su come eseguire una build, tra cui dove trovare il codice sorgente, quale ambiente di compilazione utilizzare, quali comandi di build eseguire e dove archiviare l'output della build. Un ambiente di compilazione rappresenta una combinazione di sistema operativo, linguaggio di programmazione, runtime e strumenti CodeBuild utilizzati per eseguire una build. Per ulteriori informazioni, consulta:
 - [Creazione di un progetto di compilazione](#)
 - [Riferimento ambiente di compilazione](#)
2. CodeBuild utilizza il progetto di compilazione per creare l'ambiente di compilazione.
3. CodeBuild scarica il codice sorgente nell'ambiente di compilazione e quindi utilizza le specifiche di compilazione (buildspec), come definite nel progetto di compilazione o incluse direttamente nel codice sorgente. Un buildspec è una raccolta di comandi di compilazione e impostazioni correlate, in formato YAML, che viene utilizzata per eseguire una build. CodeBuild Per ulteriori informazioni, consulta [Riferimento per buildspec](#).
4. Se esiste un output di compilazione, l'ambiente di compilazione ne carica il risultato in un bucket S3. L'ambiente di compilazione può anche eseguire le attività specificate nella buildspec (ad

esempio, l'invio di notifiche di build a un argomento di Amazon SNS). Per vedere un esempio, consulta [Esempio di notifiche di compilazione](#).

5. Mentre la build è in esecuzione, l'ambiente di compilazione invia informazioni ad CodeBuild Amazon CloudWatch Logs.
6. Mentre la build è in esecuzione, puoi utilizzare la AWS CodeBuild console o AWS SDKs ottenere informazioni riepilogative sulla build CodeBuild e informazioni dettagliate sulla build da Amazon CloudWatch Logs. AWS CLI Se usi AWS CodePipeline per eseguire build, puoi ottenere informazioni limitate sulla build da. CodePipeline

Fasi successive

Ora che ne sai di più AWS CodeBuild, ti consigliamo di seguire questi passaggi successivi:

1. Sperimenta CodeBuild in uno scenario di esempio seguendo le istruzioni riportate in [Nozioni di base utilizzando la console](#).
2. CodeBuild Utilizzalo nei tuoi scenari seguendo le istruzioni riportate in [Pianifica una compilazione](#).

Guida introduttiva con CodeBuild

Nei seguenti tutorial, viene utilizzato AWS CodeBuild per creare una raccolta di file di input di codice sorgente di esempio in una versione distribuibile del codice sorgente.

Entrambi i tutorial hanno lo stesso input e gli stessi risultati, ma uno utilizza la AWS CodeBuild console e l'altro utilizza la AWS CLI

Important

Non è consigliabile utilizzare l'account AWS root per completare questo tutorial.

Argomenti

- [Guida introduttiva all' AWS CodeBuild utilizzo della console](#)
- [Guida introduttiva all' AWS CodeBuild utilizzo di AWS CLI](#)

Guida introduttiva all' AWS CodeBuild utilizzo della console

In questo tutorial, puoi AWS CodeBuild creare una raccolta di file di input di esempio del codice sorgente (build input artefact o build input) in una versione distribuibile del codice sorgente (build output artifact o build output). In particolare, si richiede di CodeBuild utilizzare Apache Maven, uno strumento di compilazione comune, per creare un set di file di classe Java in un file Java Archive (JAR). Non è necessario avere familiarità con Apache Maven o Java per completare questa procedura guidata.

È possibile utilizzare CodeBuild tramite la CodeBuild console, il AWS CodePipeline, o il AWS CLI. AWS SDKs Questo tutorial illustra come utilizzare la CodeBuild console. Per ulteriori informazioni sull'uso di CodePipeline, consultare [Usa CodeBuild con CodePipeline](#).

Important

I passaggi di questo tutorial richiedono la creazione di risorse (ad esempio un bucket S3) che potrebbero comportare addebiti sul tuo account. AWS Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon S3 e AWS KMS CloudWatch Logs.

Per ulteriori informazioni, consulta [AWS CodeBuild i prezzi, i prezzi](#) di [Amazon S3, i prezzi](#) e [AWS Key Management Service i prezzi](#) di [Amazon CloudWatch](#) .

Argomenti

- [Fase 1: Creare il codice sorgente](#)
- [Passaggio 2: creare il file buildspec](#)
- [Fase 3: Creare due bucket S3](#)
- [Fase 4: caricamento del codice di origine e del file buildspec](#)
- [Fase 5: creazione del progetto di compilazione](#)
- [Fase 6: esecuzione della compilazione](#)
- [Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione](#)
- [Fase 8: visualizzazione di informazioni dettagliate sulla compilazione](#)
- [Fase 9: generazione dell'artefatto di output della compilazione](#)
- [Passaggio 10: eliminare i bucket S3](#)
- [Wrapping](#)

Fase 1: Creare il codice sorgente

(Parte di: [Guida introduttiva all' AWS CodeBuild utilizzo della console](#))

In questo passaggio, crei il codice sorgente che desideri CodeBuild inserire nel bucket di output. Questo codice sorgente è costituito da due file di classe Java e un Project Object Model (POM) di Apache Maven.

1. In una directory vuota sul tuo computer o istanza locale, creare la struttura della directory.

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

2. Utilizzando un editor di testo a scelta, creare questo file, nominarlo `MessageUtil.java` e quindi salvarlo nella directory `src/main/java`.

```
public class MessageUtil {
    private String message;

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

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

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

Questo file di classe crea come output la stringa di caratteri che gli viene trasmessa. Il costruttore MessageUtil imposta la stringa di caratteri. Il metodo printMessage crea l'output. Il metodo salutationMessage restituisce Hi! seguito dalla stringa di caratteri.

3. Creare questo file, nominarlo TestMessageUtil.java e quindi salvarlo nella directory /src/test/java.

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

public class TestMessageUtil {

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

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

    @Test
```

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

Questo file di classe imposta la variabile message nella classe MessageUtil su Robert. Quindi effettua dei test per vedere se l'impostazione della variabile message è riuscita controllando se le stringhe Robert e Hi! Robert sono visualizzate nell'output.

4. Creare questo file, denominarlo pom.xml e salvarlo nella directory radice (primo livello).

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

Apache Maven utilizza le istruzioni in questo file per convertire i file `MessageUtil.java` e `TestMessageUtil.java` in un file denominato `messageUtil-1.0.jar`, quindi per eseguire i test specificati.

A questo punto, la tua struttura di directory dovrebbe avere questo aspetto.

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

Passaggio 2: creare il file buildspec

(Fase precedente: [Fase 1: Creare il codice sorgente](#))

In questa fase, crei un file di specifica di compilazione. Un buildspec è una raccolta di comandi di compilazione e impostazioni correlate, in formato YAML, che viene utilizzata per eseguire una build. CodeBuild Senza una specifica di compilazione, CodeBuild non è possibile convertire con successo l'input della build in output della build o individuare l'artefatto di output della build nell'ambiente di compilazione per caricarlo nel bucket di output.

Creare questo file, denominarlo `buildspec.yml` e salvarlo nella directory radice (primo livello).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
```

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

Important

Poiché una dichiarazione della specifica di compilazione deve essere uno YAML valido, in una dichiarazione della specifica di compilazione è importante la spaziatura. Se il numero di spazi nella dichiarazione della specifica di compilazione non corrisponde a questo, la compilazione potrebbe non riuscire immediatamente. È possibile usare un convalidatore YAML per verificare se la dichiarazione della specifica di compilazione è uno YAML valido.

Note

Invece di includere un file di specifica di compilazione nel codice sorgente, è possibile dichiarare i comandi di compilazione separatamente al momento della creazione di un progetto di compilazione. Questa funzione è utile se si desidera creare il codice sorgente con diversi comandi di compilazione senza aggiornare ogni volta l'archivio del codice sorgente. Per ulteriori informazioni, consulta [Sintassi buildspec](#).

In questa dichiarazione della specifica di compilazione:

- `version` rappresenta la versione dello standard di specifica di compilazione utilizzata. Questa dichiarazione della specifica di compilazione utilizza la versione più recente, `0.2`.
- `phases` rappresenta le fasi di build durante le quali è possibile fornire istruzioni a CodeBuild per eseguire i comandi. Queste fasi di compilazione sono elencate qui come `install`, `pre_build`, `build` e `post_build`. Non puoi modificare l'ortografia dei nomi di queste fasi di compilazione e non puoi creare ulteriori nomi di fasi di compilazione.

In questo esempio, durante la `build` fase, CodeBuild esegue il comando `mvn install`. Questo comando fornisce istruzioni a Apache Maven per compilare, testare e creare pacchetti dei file di classe Java in un artefatto di output della compilazione. Per completezza, in questo esempio

alcuni comandi echo sono posizionati in ciascuna fase di build. Quando più avanti in questo tutorial visualizzerai informazioni build dettagliate, l'output di questi comandi echo può aiutarti a comprendere meglio in che modo CodeBuild esegue i comandi e in quale ordine. Anche se tutte le fasi di compilazione sono incluse in questo esempio, non è necessario includere una fase di compilazione se non prevedi di eseguire comandi durante tale fase. Per ogni fase di compilazione, CodeBuild esegue ogni comando specificato, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

- `artifacts` rappresenta l'insieme di artefatti di output della build che CodeBuild vengono caricati nel bucket di output. `files` rappresenta i file da includere nell'output della build. CodeBuild carica il singolo `messageUtil-1.0.jar` file trovato nella directory `target` relativa nell'ambiente di compilazione. Il nome del file `messageUtil-1.0.jar` e il nome della directory `target` sono basati sul modo in cui Apache Maven crea e archivia elementi di output di compilazione unicamente per questo esempio. Nelle tue build, questi nomi di file e directory sono differenti.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

A questo punto, la tua struttura di directory dovrebbe avere questo aspetto.

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

Fase 3: Creare due bucket S3

(Fase precedente: [Passaggio 2: creare il file buildspec](#))

Anche se è possibile utilizzare un singolo bucket per questo tutorial, due bucket consentono di comprendere più agevolmente la provenienza dell'input della build e la destinazione dell'output della build.

- Uno di questi bucket archivia l'input di compilazione (il bucket di input) archivia l'input della build. In questo tutorial, il nome di questo bucket di input è `codebuild-region-ID-account-ID-`

input-bucket, *region-ID* dov'è la AWS regione del bucket e *account-ID* l'ID del tuo account. AWS

- L'altro bucket (il bucket di output) archivia l'output di compilazione. In questa esercitazione, il nome di questo bucket di output è codebuild-*region-ID-account-ID*-output-bucket.

Se hai scelto nomi diversi per questi bucket, assicurati di usarli in tutto questo tutorial.

Questi due bucket devono trovarsi nella stessa AWS regione delle tue build. Ad esempio, se si richiede di CodeBuild eseguire una build nella regione Stati Uniti orientali (Ohio), anche questi bucket devono trovarsi nella regione Stati Uniti orientali (Ohio).

Per ulteriori informazioni, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

Note

Sebbene supporti CodeBuild anche gli input di compilazione memorizzati nei CodeCommit repository e Bitbucket, questo tutorial non mostra come usarli. GitHub Per ulteriori informazioni, consulta [Pianifica una compilazione](#).

Fase 4: caricamento del codice di origine e del file buildspec

(Fase precedente: [Fase 3: Creare due bucket S3](#))

In questa fase, aggiungi il codice sorgente e il file di specifica di compilazione al bucket di input.

Utilizzando l'utilità zip del tuo sistema operativo, crea un file denominato MessageUtil.zip che includa MessageUtil.java, TestMessageUtil.java, pom.xml e buildspec.yml.

La struttura di directory del file MessageUtil.zip deve avere il seguente aspetto.

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



```
`-- TestMessageUtil.java
```

⚠ Important

Non includere la directory (*root directory name*), ma solo le directory e i file contenuti nella directory (*root directory name*).

Carica il file `MessageUtil.zip` nel bucket di input denominato `codebuild-region-ID-account-ID-input-bucket`.

⚠ Important

Per i repository Bitbucket CodeCommit GitHub, per convenzione, è necessario memorizzare un file delle specifiche di compilazione denominato `buildspec.yml` nella radice (livello superiore) di ciascun repository o includere la dichiarazione delle specifiche di compilazione come parte della definizione del progetto di compilazione. Non creare un file ZIP che contiene il codice sorgente dell'archivio e il file della specifica di compilazione.

Solo per gli input di compilazione archiviati nei bucket S3, è necessario creare un file ZIP che contiene il codice di origine e, per convenzione, un file della specifica di compilazione denominato `buildspec.yml` nel root (livello superiore) o includere la dichiarazione della specifica di compilazione come parte della definizione del progetto di compilazione.

Se si desidera utilizzare un nome diverso per il file della specifica di compilazione, oppure se si desidera fare riferimento a una specifica di compilazione in un percorso diverso da quello principale, è possibile specificare una sostituzione della specifica di compilazione come parte della definizione del progetto di compilazione. Per ulteriori informazioni, consulta [Nome del file buildspec e posizione di storage](#).

Fase 5: creazione del progetto di compilazione

(Fase precedente: [Fase 4: caricamento del codice di origine e del file buildspec](#))

In questo passaggio, crei un progetto di compilazione che AWS CodeBuild utilizza per eseguire la build. Un progetto di compilazione include informazioni su come eseguire una build, tra cui dove trovare il codice sorgente, quale ambiente di compilazione utilizzare, quali comandi di compilazione eseguire e dove archiviare l'output della build. Un ambiente di compilazione rappresenta una combinazione di sistema operativo, linguaggio di programmazione, runtime e strumenti CodeBuild

utilizzati per eseguire una build. L'ambiente di compilazione è espresso come immagine Docker. Per ulteriori informazioni, consulta l'argomento relativo alla [panoramica del Docker](#) sul sito web Docker Docs.

Per questo ambiente di compilazione, si richiede di CodeBuild utilizzare un'immagine Docker che contenga una versione del Java Development Kit (JDK) e Apache Maven.

Per creare il progetto build

1. [Accedi e apri la console su codebuild/home Console di gestione AWS . AWS CodeBuild https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Usa il selettore di AWS regione per scegliere una AWS regione in cui è supportata. CodeBuild Per ulteriori informazioni, consulta [Endpoint e quote AWS CodeBuild](#) nella Riferimenti generali di Amazon Web Services.
3. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
4. Nella pagina Create build project (Crea progetto di compilazione), in Project Configuration (Configurazione progetto), per Project name (Nome progetto), immetti un nome per questo progetto di compilazione (in questo esempio, codebuild-demo-project). I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Se scegli un nome differente, assicurati di utilizzarlo in tutto il tutorial.

Note

Nella pagina Create build project (Crea progetto di compilazione), può comparire un messaggio di errore simile al seguente: You are not authorized to perform this operation (Non sei autorizzato a eseguire questa operazione). Ciò è probabilmente dovuto al fatto che hai effettuato l'accesso Console di gestione AWS come utente che non dispone delle autorizzazioni per creare un progetto di compilazione. Per risolvere questo problema, esci da Console di gestione AWS, quindi accedi nuovamente con le credenziali appartenenti a una delle seguenti entità IAM:

- Un utente amministratore nel tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con le AWSCodeBuildAdminAccess politiche IAMFullAccess gestite collegate a quell'utente o a un gruppo IAM a cui l'utente

appartiene. AmazonS3ReadOnlyAccess Se nel tuo AWS account non hai un utente o un gruppo con queste autorizzazioni e non riesci ad aggiungere queste autorizzazioni al tuo utente o gruppo, contatta l'amministratore dell' AWS account per ricevere assistenza. Per ulteriori informazioni, consulta [AWS politiche gestite \(predefinite\) per AWS CodeBuild](#).

Entrambe le opzioni includono le autorizzazioni di amministratore che consentono di creare un progetto di compilazione in modo da poter completare questo tutorial. È consigliabile utilizzare sempre le autorizzazioni minime necessarie per eseguire l'attività. Per ulteriori informazioni, consulta [AWS CodeBuild riferimento alle autorizzazioni](#).

5. In Source, per provider di origine, scegli Amazon S3.
6. Per Bucket, scegli codebuild- - **region-ID** -input-bucket. **account-ID**
7. Per S3 object key (Chiave oggetto S3), immettere **MessageUtil.zip**.
8. In Environment (Ambiente), per Environment image (Immagine ambiente), lasciare selezionato Managed image (Immagine gestita).
9. Per il sistema operativo, scegli Amazon Linux.
10. In Runtime(s) (Runtime), seleziona Standard.
11. Per Image, scegli aws/codebuild/amazonlinux-x86_64-standard:corretto11.
12. Alla voce Service role (Ruolo del servizio), lasciare selezionato New service role (Nuovo ruolo del servizio) e lasciare immutata la voce Role name (Nome ruolo).
13. Per Buildspec (Specifica di compilazione), lasciare selezionata l'opzione Use a buildspec file (Utilizza un file buildspec).
14. In Artifacts, per Tipo, scegli Amazon S3.
15. Per il nome del bucket, scegli codebuild - - -output-bucket. **region-ID account-ID**
16. Lasciare vuoti i campi Name (Nome) e Path (Percorso).
17. Scegliere Create build project (Crea progetto di compilazione).

Fase 6: esecuzione della compilazione

(Fase precedente: [Fase 5: creazione del progetto di compilazione](#))

In questo passaggio, si ordina di eseguire la build con AWS CodeBuild le impostazioni del progetto di compilazione.

Per eseguire la build

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Nell'elenco dei progetti di compilazione, scegli codebuild-demo-project, quindi scegli Avvia compilazione. La compilazione viene avviata immediatamente.

Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione

(Fase precedente: [Fase 6: esecuzione della compilazione](#))

In questa fase, visualizzi un riepilogo delle informazioni sullo stato della compilazione.

Per visualizzare un riepilogo delle informazioni sulla compilazione

1. Se la *<build-ID>* pagina codebuild-demo-project: non è visualizzata, nella barra di navigazione, scegli Cronologia build. Quindi, nell'elenco dei progetti di compilazione, per Project, scegli il link Build run for codebuild-demo-project. Deve esserci un solo collegamento corrispondente. Se è stata completato questo tutorial in precedenza, scegliere il collegamento con il valore più recente nella colonna Completed (Completato).
2. Nella pagina di stato della compilazione, nei dettagli della fase, devono essere visualizzate le seguenti fasi di compilazione, con Succeeded nella colonna Stato:
 - SUBMITTED (INVIATO)
 - IN CODA
 - PROVISIONING
 - DOWNLOAD_SOURCE
 - INSTALL (INSTALLA)
 - PRE_BUILD
 - BUILD
 - POST_BUILD
 - UPLOAD_ARTIFACTS (CARICA_ARTEFATTO)
 - FINALIZING (FINALIZZAZIONE)
 - COMPLETED

In Build Status (Stato della compilazione), deve essere visualizzato Succeeded (Riuscito).

Se invece viene visualizzato In Progress (In corso), scegliere il pulsante di aggiornamento.

3. Accanto a ogni fase di build, il valore Duration (Durata) indica la durata della fase di build. Il valore End time (Ora fine) indica il momento in cui è terminata quella fase di build.

Fase 8: visualizzazione di informazioni dettagliate sulla compilazione

(Fase precedente: [Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione](#))

In questo passaggio, visualizzi informazioni dettagliate sulla build in CloudWatch Logs.

Note

Per proteggere le informazioni riservate, nei CodeBuild log sono nascoste le seguenti informazioni:

- AWS chiave IDs di accesso. Per ulteriori informazioni, consulta [la sezione Gestione delle chiavi di accesso per gli utenti IAM](#) nella Guida per l'AWS Identity and Access Management utente.
- Stringhe specificate utilizzando l'archivio parametri. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.
- Stringhe specificate utilizzando. Gestione dei segreti AWS Per ulteriori informazioni, consulta [Gestione delle chiavi](#).

Per visualizzare informazioni dettagliate sulla compilazione

1. Con la pagina dei dettagli della compilazione ancora visualizzata dalla fase precedente, le ultime 10.000 righe del log di compilazione vengono visualizzate in Build logs (Log di compilazione). Per visualizzare l'intero registro di compilazione nei CloudWatch registri, scegli il link Visualizza l'intero registro.
2. Nel flusso di CloudWatch log di Logs, puoi sfogliare gli eventi di registro. Per impostazione predefinita, viene visualizzato solo l'ultimo set di eventi di log. Per visualizzare gli eventi di log precedenti, scorrere all'inizio dell'elenco.

3. In questo tutorial, la maggior parte degli eventi di log contengono informazioni dettagliate su come CodeBuild scarica e installa i file delle dipendenze di compilazione nel proprio ambiente di compilazione, cosa che probabilmente non interessa. È possibile utilizzare la casella Filter events (Filtra eventi) per ridurre le informazioni visualizzate. Ad esempio, se si immette "[INFO]" in Filter events (Filtra eventi) vengono visualizzati solo gli eventi contenenti [INFO]. Per ulteriori informazioni, consulta [Filter and pattern syntax](#) nella Amazon CloudWatch User Guide.

Fase 9: generazione dell'artefatto di output della compilazione

(Fase precedente: [Fase 8: visualizzazione di informazioni dettagliate sulla compilazione](#))

In questo passaggio, ottieni il `messageUtil-1.0.jar` file CodeBuild creato e caricato nel bucket di output.

Puoi utilizzare la CodeBuild console o la console Amazon S3 per completare questo passaggio.

Per ottenere l'artefatto di output della build (console)AWS CodeBuild

1. Con la CodeBuild console ancora aperta e la pagina dei dettagli della build ancora visualizzata nel passaggio precedente, scegli la scheda Dettagli della build e scorri verso il basso fino alla sezione Artefatti.

Note

Se la pagina dei dettagli della build non è visualizzata, nella barra di navigazione, scegli Cronologia build, quindi scegli il link Build run.

2. Il link alla cartella Amazon S3 si trova nella posizione di caricamento degli artefatti. Questo collegamento apre la cartella in Amazon S3 in cui si trova il file degli artefatti di output della `messageUtil-1.0.jar` build.

Per ottenere l'elemento di output della build (console Amazon S3)

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. Aprire `codebuild-region-ID-account-ID-output-bucket`.
3. Aprire la cartella `codebuild-demo-project`.

4. Aprire la cartella `target`, contenente il file dell'artefatto di output di compilazione `messageUtil-1.0.jar`.

Passaggio 10: eliminare i bucket S3

(Fase precedente: [Fase 9: generazione dell'artefatto di output della compilazione](#))

Per evitare addebiti continui AWS sul tuo account, puoi eliminare i bucket di input e output utilizzati in questo tutorial. Per istruzioni, consulta [Eliminare o svuotare un bucket nella Guida per l'utente](#) di Amazon Simple Storage Service.

Se utilizzi l'utente IAM o un utente IAM amministratore per eliminare questi bucket, l'utente deve disporre di più autorizzazioni di accesso. Aggiungi la seguente istruzione tra i marker (**### BEGIN ADDING STATEMENT HERE ###e### END ADDING STATEMENTS HERE ###**) a una politica di accesso esistente per l'utente.

Le ellissi (...) in questa affermazione sono usate per brevità. Non rimuovere nessuna istruzione nella policy di accesso esistente. Non immettere queste ellissi nella policy.

JSON

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

Wrapping

In questo tutorial, creavi AWS CodeBuild un set di file di classe Java in un file JAR. Quindi hai visualizzato i risultati della compilazione.

Ora puoi provare a utilizzarlo CodeBuild nei tuoi scenari. Segui le istruzioni in [Pianifica una compilazione](#). Se non ti senti ancora pronto, puoi provare a compilare alcuni degli esempi. Per ulteriori informazioni, consulta [Usa esempi basati su casi per CodeBuild](#).

Guida introduttiva all' AWS CodeBuild utilizzo di AWS CLI

In questo tutorial, si utilizza AWS CodeBuild per creare una raccolta di file di input di codice sorgente di esempio (chiamati build input artifacts o build input) in una versione distribuibile del codice sorgente (chiamata build output artifact o build output). In particolare, si richiede di CodeBuild utilizzare Apache Maven, uno strumento di compilazione comune, per creare un set di file di classe Java in un file Java Archive (JAR). Non è necessario avere familiarità con Apache Maven o Java per completare questa procedura guidata.

È possibile utilizzare CodeBuild tramite la CodeBuild console, il AWS CodePipeline, o il AWS CLI. AWS SDKs Questo tutorial mostra come utilizzare CodeBuild con. AWS CLI Per informazioni sull'utilizzo CodePipeline, vedere [Usa CodeBuild con CodePipeline](#).

Important

I passaggi di questo tutorial richiedono la creazione di risorse (ad esempio un bucket S3) che potrebbero comportare addebiti sul tuo AWS account. Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon S3 e AWS KMS CloudWatch Logs. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i prezzi di Amazon S3, i prezzi e AWS Key Management Service i prezzi di Amazon CloudWatch](#).

Argomenti

- [Fase 1: Creare il codice sorgente](#)
- [Passaggio 2: creare il file buildspec](#)
- [Fase 3: Creare due bucket S3](#)
- [Fase 4: caricamento del codice di origine e del file buildspec](#)
- [Fase 5: creazione del progetto di compilazione](#)
- [Fase 6: esecuzione della compilazione](#)
- [Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione](#)

- [Fase 8: visualizzazione di informazioni dettagliate sulla compilazione](#)
- [Fase 9: generazione dell'artefatto di output della compilazione](#)
- [Passaggio 10: eliminare i bucket S3](#)
- [Wrapping](#)

Fase 1: Creare il codice sorgente

(Parte di: [Guida introduttiva all' AWS CodeBuild utilizzo di AWS CLI](#))

In questo passaggio, crei il codice sorgente che desideri CodeBuild inserire nel bucket di output. Questo codice sorgente è costituito da due file di classe Java e un Project Object Model (POM) di Apache Maven.

1. In una directory vuota sul tuo computer o istanza locale, creare la struttura della directory.

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

2. Utilizzando un editor di testo a scelta, creare questo file, nominarlo `MessageUtil.java` e quindi salvarlo nella directory `src/main/java`.

```
public class MessageUtil {
    private String message;

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

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

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

```
    return message;
  }
}
```

Questo file di classe crea come output la stringa di caratteri che gli viene trasmessa. Il costruttore `MessageUtil` imposta la stringa di caratteri. Il metodo `printMessage` crea l'output. Il metodo `salutationMessage` restituisce `Hi!` seguito dalla stringa di caratteri.

3. Creare questo file, nominarlo `TestMessageUtil.java` e quindi salvarlo nella directory `/src/test/java`.

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

public class TestMessageUtil {

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

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

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

Questo file di classe imposta la variabile `message` nella classe `MessageUtil` su `Robert`. Quindi effettua dei test per vedere se l'impostazione della variabile `message` è riuscita controllando se le stringhe `Robert` e `Hi! Robert` sono visualizzate nell'output.

4. Creare questo file, denominarlo `pom.xml` e salvarlo nella directory radice (primo livello).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

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

Apache Maven utilizza le istruzioni in questo file per convertire i file `MessageUtil.java` e `TestMessageUtil.java` in un file denominato `messageUtil-1.0.jar`, quindi per eseguire i test specificati.

A questo punto, la tua struttura di directory dovrebbe avere questo aspetto.

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

```
`-- TestMessageUtil.java
```

Passaggio 2: creare il file buildspec

(Fase precedente: [Fase 1: Creare il codice sorgente](#))

In questa fase, crei un file di specifica di compilazione. Un buildspec è una raccolta di comandi di compilazione e impostazioni correlate, in formato YAML, che viene utilizzata per eseguire una build. CodeBuild Senza una specifica di compilazione, CodeBuild non è possibile convertire con successo l'input della build in output della build o individuare l'artefatto di output della build nell'ambiente di compilazione per caricarlo nel bucket di output.

Creare questo file, denominarlo `buildspec.yml` e salvarlo nella directory radice (primo livello).

```
version: 0.2

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

Important

Poiché una dichiarazione della specifica di compilazione deve essere uno YAML valido, in una dichiarazione della specifica di compilazione è importante la spaziatura. Se il numero di spazi nella dichiarazione della specifica di compilazione non corrisponde a questo, la

compilazione potrebbe non riuscire immediatamente. È possibile usare un convalidatore YAML per verificare se la dichiarazione della specifica di compilazione è uno YAML valido.

Note

Invece di includere un file di specifica di compilazione nel codice sorgente, è possibile dichiarare i comandi di compilazione separatamente al momento della creazione di un progetto di compilazione. Questa funzione è utile se si desidera creare il codice sorgente con diversi comandi di compilazione senza aggiornare ogni volta l'archivio del codice sorgente. Per ulteriori informazioni, consulta [Sintassi buildspec](#).

In questa dichiarazione della specifica di compilazione:

- `version` rappresenta la versione dello standard di specifica di compilazione utilizzata. Questa dichiarazione della specifica di compilazione utilizza la versione più recente, `0.2`.
- `phases` rappresenta le fasi di build durante le quali è possibile fornire istruzioni a CodeBuild per eseguire i comandi. Queste fasi di compilazione sono elencate qui come `install`, `pre_build`, `build` e `post_build`. Non puoi modificare l'ortografia dei nomi di queste fasi di compilazione e non puoi creare ulteriori nomi di fasi di compilazione.

In questo esempio, durante la `build` fase, CodeBuild esegue il comando `mvn install`. Questo comando fornisce istruzioni a Apache Maven per compilare, testare e creare pacchetti dei file di classe Java in un artefatto di output della compilazione. Per completezza, in questo esempio alcuni comandi `echo` sono posizionati in ciascuna fase di build. Quando più avanti in questo tutorial visualizzerai informazioni build dettagliate, l'output di questi comandi `echo` può aiutarti a comprendere meglio in che modo CodeBuild esegue i comandi e in quale ordine. Anche se tutte le fasi di compilazione sono incluse in questo esempio, non è necessario includere una fase di compilazione se non prevedi di eseguire comandi durante tale fase. Per ogni fase di compilazione, CodeBuild esegue ogni comando specificato, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

- `artifacts` rappresenta l'insieme di artefatti di output della build che CodeBuild vengono caricati nel bucket di output. `files` rappresenta i file da includere nell'output della build. CodeBuild carica il singolo file `messageUtil-1.0.jar` trovato nella directory `target` relativa nell'ambiente di compilazione. Il nome del file `messageUtil-1.0.jar` e il nome della directory `target` sono basati sul modo in cui Apache Maven crea e archivia elementi di output di compilazione unicamente per questo esempio. Nelle tue build, questi nomi di file e directory sono differenti.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

A questo punto, la tua struttura di directory dovrebbe avere questo aspetto.

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

Fase 3: Creare due bucket S3

(Fase precedente: [Passaggio 2: creare il file buildspec](#))

Anche se è possibile utilizzare un singolo bucket per questo tutorial, due bucket consentono di comprendere più agevolmente la provenienza dell'input della build e la destinazione dell'output della build.

- Uno di questi bucket archivia l'input di compilazione (il bucket di input) archivia l'input della build. In questo tutorial, il nome di questo bucket di input è `codebuild-region-ID-account-ID-input-bucket`, *region-ID* dov'è la AWS regione del bucket e *account-ID* l'ID del tuo account. AWS
- L'altro bucket (il bucket di output) archivia l'output di compilazione. In questa esercitazione, il nome di questo bucket di output è `codebuild-region-ID-account-ID-output-bucket`.

Se hai scelto nomi diversi per questi bucket, assicurati di usarli in tutto questo tutorial.

Questi due bucket devono trovarsi nella stessa AWS regione delle tue build. Ad esempio, se si richiede di CodeBuild eseguire una build nella regione Stati Uniti orientali (Ohio), anche questi bucket devono trovarsi nella regione Stati Uniti orientali (Ohio).

Per ulteriori informazioni, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

 Note

Sebbene supporti CodeBuild anche gli input di compilazione memorizzati nei CodeCommit repository e Bitbucket, questo tutorial non mostra come usarli. GitHub Per ulteriori informazioni, consulta [Pianifica una compilazione](#).

Fase 4: caricamento del codice di origine e del file buildspec

(Fase precedente: [Fase 3: Creare due bucket S3](#))

In questa fase, aggiungi il codice sorgente e il file di specifica di compilazione al bucket di input.

Utilizzando l'utilità zip del tuo sistema operativo, crea un file denominato `MessageUtil.zip` che includa `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` e `buildspec.yml`.

La struttura di directory del file `MessageUtil.zip` deve avere il seguente aspetto.

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

 Important

Non includere la directory (*root directory name*), ma solo le directory e i file contenuti nella directory (*root directory name*).

Carica il file `MessageUtil.zip` nel bucket di input denominato `codebuild-region-ID-account-ID-input-bucket`.

Important

Per i repository Bitbucket CodeCommit GitHub, per convenzione, è necessario memorizzare un file delle specifiche di compilazione denominato `buildspec.yml` nella radice (livello superiore) di ciascun repository o includere la dichiarazione delle specifiche di compilazione come parte della definizione del progetto di compilazione. Non creare un file ZIP che contiene il codice sorgente dell'archivio e il file della specifica di compilazione.

Solo per gli input di compilazione archiviati nei bucket S3, è necessario creare un file ZIP che contiene il codice di origine e, per convenzione, un file della specifica di compilazione denominato `buildspec.yml` nel root (livello superiore) o includere la dichiarazione della specifica di compilazione come parte della definizione del progetto di compilazione.

Se si desidera utilizzare un nome diverso per il file della specifica di compilazione, oppure se si desidera fare riferimento a una specifica di compilazione in un percorso diverso da quello principale, è possibile specificare una sostituzione della specifica di compilazione come parte della definizione del progetto di compilazione. Per ulteriori informazioni, consulta [Nome del file buildspec e posizione di storage](#).

Fase 5: creazione del progetto di compilazione

(Fase precedente: [Fase 4: caricamento del codice di origine e del file buildspec](#))

In questo passaggio, crei un progetto di compilazione che AWS CodeBuild utilizza per eseguire la build. Un progetto di compilazione include informazioni su come eseguire una build, tra cui dove trovare il codice sorgente, quale ambiente di compilazione utilizzare, quali comandi di compilazione eseguire e dove archiviare l'output della build. Un ambiente di compilazione rappresenta una combinazione di sistema operativo, linguaggio di programmazione, runtime e strumenti CodeBuild utilizzati per eseguire una build. L'ambiente di compilazione è espresso come immagine Docker. Per ulteriori informazioni, consulta l'argomento relativo alla [panoramica del Docker](#) sul sito web Docker Docs.

Per questo ambiente di compilazione, si richiede di CodeBuild utilizzare un'immagine Docker che contenga una versione del Java Development Kit (JDK) e Apache Maven.

Per creare il progetto build

1. Usa il per eseguire il comando: AWS CLI create-project


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

Nell'output vengono visualizzati dati in formato JSON. Copia i dati in un file denominato `create-project.json` in una posizione sul computer o sull'istanza locale in cui AWS CLI è installato. Se si sceglie di utilizzare un nome di file diverso, assicurarsi di utilizzarlo durante tutto il tutorial.

Modificare i dati copiati per seguire questo formato, quindi salvare i risultati:

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

Sostituisci *serviceIAMRole* con l'Amazon Resource Name (ARN) di un ruolo di CodeBuild servizio (ad esempio, `arn:aws:iam::account-ID:role/role-name`). Per crearne uno, consulta [Consenti CodeBuild di interagire con altri servizi AWS](#).

In questi dati:

- `name` rappresenta un identificatore necessario per questo progetto di compilazione (in questo esempio, `codebuild-demo-project`). I nomi dei progetti di compilazione devono essere univoci per tutti i progetti di compilazione nel tuo account.
- `source`, `type` è un valore obbligatorio che rappresenta il tipo di repository del codice sorgente (in questo esempio, `S3` per un bucket Amazon S3).
- Per `source`, `location` rappresenta il percorso per il codice sorgente (in questo esempio, il nome del bucket di input seguito dal nome del file ZIP).

- `Forartifacts`, `type` è un valore obbligatorio che rappresenta il tipo di repository dell'elemento di output della build (in questo esempio, S3 per un bucket Amazon S3).
- Per `artifacts`, `location` rappresenta il nome del bucket di output creato o identificato in precedenza (in questo esempio, `codebuild-region-ID-account-ID-output-bucket`).
- `Forenvironment`, `type` è un valore obbligatorio che rappresenta il tipo di ambiente di compilazione (in questo esempio, `LINUX_CONTAINER`).
- `Forenvironment`, `image` è un valore obbligatorio che rappresenta il nome dell'immagine Docker e la combinazione di tag utilizzata da questo progetto di build, come specificato dal tipo di repository di immagini Docker (in questo esempio, `aws/codebuild/standard:5.0` per un'immagine Docker nel repository di immagini CodeBuild Docker). `aws/codebuild/standard` è il nome dell'immagine Docker. `5.0` è il tag dell'immagine Docker.

Per trovare più immagini Docker da utilizzare negli scenari, consulta il [Riferimento ambiente di compilazione](#).

- `Forenvironment`, `computeType` è un valore obbligatorio che rappresenta le risorse di elaborazione CodeBuild utilizzate (in questo esempio, `BUILD_GENERAL1_SMALL`).

Note

Altri valori disponibili nel formato JSON originale dei dati, ad esempio `description`, `buildspec`, `auth` (inclusi `type` e `resource`), `path`, `namespaceType`, `name` (per `artifacts`), `packaging`, `environmentVariables` (inclusi `name` e `value`), `timeoutInMinutes`, `encryptionKey` e `tags` (inclusi `key` e `value`) sono facoltativi. Non vengono utilizzati in questo tutorial, perciò non vengono mostrati qui. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).

2. Passare alla directory contenente il file appena salvato, quindi eseguire nuovamente il comando `create-project`.

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

Se l'operazione ha esito positivo, dati simili a questo appaiono nell'output.

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

```
{
  "serviceRole": "serviceIAMRole",
  "tags": [],
  "artifacts": {
    "packaging": "NONE",
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "name": "message-util.zip"
  },
  "lastModified": 1472661575.244,
  "timeoutInMinutes": 60,
  "created": 1472661575.244,
  "environment": {
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/standard:5.0",
    "type": "LINUX_CONTAINER",
    "environmentVariables": []
  },
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
  "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
}
```

- `project` rappresenta le informazioni su questo progetto di compilazione.
- `tags` rappresenta tutti i tag dichiarati.
- `packaging` rappresenta il modo in cui l'elemento di output di compilazione viene archiviato nel bucket di output. `NONE` significa che una cartella viene creata nel bucket di output. L'elemento di output di compilazione viene archiviato in questa cartella.
- `lastModified` rappresenta l'orario, in formato Unix, in cui le informazioni sul progetto di compilazione sono state modificate l'ultima volta.
- `timeoutInMinutes` rappresenta il numero di minuti trascorsi i quali CodeBuild interrompe la compilazione se la compilazione non è stata completata. Il valore predefinito è 60 minuti.
- `created` rappresenta l'orario, in formato Unix, in cui il progetto di compilazione è stato creato.
- `environmentVariables` rappresenta tutte le variabili di ambiente dichiarate e disponibili CodeBuild per l'uso durante la compilazione.

- `encryptionKey` rappresenta l'ARN della chiave gestita dal cliente CodeBuild utilizzata per crittografare l'artefatto di output della build.
- `arn` rappresenta l'ARN del progetto di compilazione.

Note

Dopo aver eseguito il `create-project` comando, potrebbe essere emesso un messaggio di errore simile al seguente: Utente: non autorizzato a eseguire: **`user-ARN`** codebuild:. CreateProject Ciò è probabilmente dovuto al fatto che lo hai configurato AWS CLI con le credenziali di un utente che non dispone di autorizzazioni sufficienti da utilizzare per CodeBuild creare progetti di compilazione. Per risolvere questo problema, configura AWS CLI con credenziali appartenenti a una delle seguenti entità IAM:

- Un utente amministratore nel tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con le `AWSCodeBuildAdminAccess` politiche `IAMFullAccess` gestite collegate a quell'utente o a un gruppo IAM a cui l'utente appartiene. `AmazonS3ReadOnlyAccess` Se nel tuo AWS account non hai un utente o un gruppo con queste autorizzazioni e non riesci ad aggiungere queste autorizzazioni al tuo utente o gruppo, contatta l'amministratore dell' AWS account per ricevere assistenza. Per ulteriori informazioni, consulta [AWS politiche gestite \(predefinite\) per AWS CodeBuild](#).

Fase 6: esecuzione della compilazione

(Fase precedente: [Fase 5: creazione del progetto di compilazione](#))

In questo passaggio, si ordina di AWS CodeBuild eseguire la build con le impostazioni del progetto di compilazione.

Per eseguire la build

1. Usa il AWS CLI per eseguire il `start-build` comando:

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

Sostituiscilo *project-name* con il nome del progetto di build del passaggio precedente (ad esempio,codebuild-demo-project).

2. Se eseguito correttamente, nell'output compaiono dei dati simili ai seguenti:

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

- build rappresenta le informazioni su questa compilazione.
- buildComplete indica se la compilazione è stata completata (true). In caso contrario, false.
- initiator rappresenta l'entità che ha avviato la compilazione.
- artifacts rappresenta le informazioni sull'output di compilazione, tra cui la posizione.
- projectName rappresenta il nome del progetto di compilazione.

- `buildStatus` rappresenta l'attuale stato della compilazione nel momento in cui il comando `start-build` è stato eseguito.
- `currentPhase` rappresenta la fase di compilazione attuale nel momento in cui il comando `start-build` è stato eseguito.
- `startTime` rappresenta l'orario, in formato Unix, in cui il processo di compilazione è stato avviato.
- `id` rappresenta l'ID della compilazione.
- `arn` rappresenta l'ARN della compilazione.

Annotare il valore `id`. Questo valore servirà nella fase successiva.

Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione

(Fase precedente: [Fase 6: esecuzione della compilazione](#))

In questa fase, visualizzi un riepilogo delle informazioni sullo stato della compilazione.

Per visualizzare un riepilogo delle informazioni sulla compilazione

- Usa il AWS CLI per eseguire il `batch-get-builds` comando.

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

Sostituisci *id* con il `id` valore visualizzato nell'output del passaggio precedente.

Se l'operazione ha esito positivo, dati simili a questo appaiono nell'output.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
```

```

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

```

```
}
```

- `buildsNotFound` rappresenta la build IDs per tutte le build in cui le informazioni non sono disponibili. In questo esempio, deve essere vuoto.
- `builds` rappresenta le informazioni su ogni build di sono disponibili informazioni. In questo esempio, nell'output appaiono informazioni relative a una sola compilazione.
- `phases` rappresenta il set di fasi di compilazione che CodeBuild esegue durante il processo di compilazione. Le informazioni su ogni fase di compilazione sono elencate separatamente come `startTime`, `endTime` e `durationInSeconds` (l'orario in cui la fase di compilazione è stata avviata ed è terminata, espresso in formato Unix, e della sua durata in secondi), ma anche `phaseType`, ad esempio (SUBMITTED, PROVISIONING, DOWNLOAD_SOURCE, INSTALL, PRE_BUILD, BUILD, POST_BUILD, UPLOAD_ARTIFACTS, FINALIZING o COMPLETED) e `phaseStatus` (ad esempio, SUCCEEDED, FAILED, FAULT, TIMED_OUT, IN_PROGRESS o STOPPED). La prima volta che esegui il comando `batch-get-builds`, potrebbero esserci poche fasi (o nessuna). Dopo esecuzioni successive del comando `batch-get-builds` con lo stesso ID di compilazione, nell'output dovrebbero essere visualizzate più fasi di compilazione.
- `logs` rappresenta informazioni in Amazon CloudWatch Logs sui log della build.
- `md5sum` MD5 e `sha256sum` rappresentano gli hash SHA-256 dell'artefatto di output della build. Questi appaiono nell'output solo se il valore `packaging` del progetto di compilazione è impostato su ZIP. Non hai impostato tale valore in questo tutorial. Puoi utilizzare questi hash insieme a uno strumento di checksum per la conferma sia dell'integrità sia dell'autenticità dei file.

Note

Puoi anche utilizzare la console Amazon S3 per visualizzare questi hash. Seleziona la casella accanto all'artefatto di output della compilazione, scegli **Actions** (Operazioni), quindi **Properties** (Proprietà). Nel riquadro **Proprietà**, espandi **Metadati** e visualizza i valori per `x-amz-meta-codebuild-content-md5` e `-content-sha256`. `x-amz-meta-codebuild` (Nella console Amazon S3, il `ETag` valore dell'artefatto di output della build non deve essere interpretato come l'hash MD5 o SHA-256.)
Se usi il AWS SDKs per ottenere questi hash, i valori sono denominati `e.codebuild-content-md5` e `e.codebuild-content-sha256`.

- `endTime` rappresenta l'orario, in formato Unix, in cui il processo di compilazione è terminato.

 Note

I metadati di Amazon S3 hanno un' CodeBuild intestazione denominata `x-amz-meta-codebuild-buildarn` che contiene la CodeBuild build che pubblica gli `buildArn` artefatti su Amazon S3. `buildArn` viene aggiunto per consentire il tracciamento della fonte per le notifiche e per fare riferimento alla build da cui viene generato l'artefatto.

Fase 8: visualizzazione di informazioni dettagliate sulla compilazione

(Fase precedente: [Fase 7: visualizzazione del riepilogo delle informazioni sulla compilazione](#))

In questo passaggio, visualizzerai informazioni dettagliate sulla tua build in CloudWatch Logs.

 Note

Per proteggere le informazioni riservate, nei CodeBuild log sono nascoste le seguenti informazioni:

- AWS chiave IDs di accesso. Per ulteriori informazioni, consulta [la sezione Gestione delle chiavi di accesso per gli utenti IAM](#) nella Guida per l'AWS Identity and Access Management utente.
- Stringhe specificate utilizzando l'archivio parametri. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.
- Stringhe specificate utilizzando. Gestione dei segreti AWS Per ulteriori informazioni, consulta [Gestione delle chiavi](#).

Per visualizzare informazioni dettagliate sulla compilazione

1. Utilizzare il browser Web per accedere alla posizione `deepLink` apparsa nell'output nella fase precedente (ad esempio, `https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).

2. Nel flusso di CloudWatch log dei registri, è possibile sfogliare gli eventi del registro. Per impostazione predefinita, viene visualizzato solo l'ultimo set di eventi di log. Per visualizzare gli eventi di log precedenti, scorrere all'inizio dell'elenco.
3. In questo tutorial, la maggior parte degli eventi di log contengono informazioni dettagliate su come CodeBuild scarica e installa i file delle dipendenze di compilazione nel proprio ambiente di compilazione, cosa che probabilmente non interessa. È possibile utilizzare la casella Filter events (Filtra eventi) per ridurre le informazioni visualizzate. Ad esempio, se si immette "[INFO]" in Filter events (Filtra eventi) vengono visualizzati solo gli eventi contenenti [INFO]. Per ulteriori informazioni, consulta [Filter and pattern syntax](#) nella Amazon CloudWatch User Guide.

Queste parti di un flusso di log di CloudWatch Logs si riferiscono a questo tutorial.

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

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

In questo esempio, sono state completate CodeBuild con successo le fasi di pre-compilazione, compilazione e post-compilazione. Ha eseguito lo unit test e la creazione del file `messageUtil-1.0.jar` è riuscita.

Fase 9: generazione dell'artefatto di output della compilazione

(Fase precedente: [Fase 8: visualizzazione di informazioni dettagliate sulla compilazione](#))

In questo passaggio, si ottiene il `messageUtil-1.0.jar` file CodeBuild creato e caricato nel bucket di output.

Puoi utilizzare la CodeBuild console o la console Amazon S3 per completare questo passaggio.

Per ottenere l'artefatto di output della build (console)AWS CodeBuild

1. Con la CodeBuild console ancora aperta e la pagina dei dettagli della build ancora visualizzata nel passaggio precedente, scegli la scheda Dettagli della build e scorri verso il basso fino alla sezione Artefatti.

 Note

Se la pagina dei dettagli della build non è visualizzata, nella barra di navigazione, scegli Cronologia build, quindi scegli il link Build run.

2. Il link alla cartella Amazon S3 si trova nella posizione di caricamento degli artefatti. Questo collegamento apre la cartella in Amazon S3 in cui si trova il file degli artefatti di output della messageUtil-1.0.jar build.

Per ottenere l'elemento di output della build (console Amazon S3)

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. Aprire codebuild-*region-ID-account-ID*-output-bucket.
3. Aprire la cartella codebuild-demo-project.
4. Aprire la cartella target, contenente il file dell'artefatto di output di compilazione messageUtil-1.0.jar.

Passaggio 10: eliminare i bucket S3

(Fase precedente: [Fase 9: generazione dell'artefatto di output della compilazione](#))

Per evitare addebiti continui AWS sul tuo account, puoi eliminare i bucket di input e output utilizzati in questo tutorial. Per istruzioni, consulta [Eliminare o svuotare un bucket nella Guida per l'utente](#) di Amazon Simple Storage Service.

Se utilizzi l'utente IAM o un utente IAM amministratore per eliminare questi bucket, l'utente deve disporre di più autorizzazioni di accesso. Aggiungi la seguente istruzione tra i marker (**### BEGIN ADDING STATEMENT HERE ###e### END ADDING STATEMENTS HERE ###**) a una politica di accesso esistente per l'utente.

Le ellissi (...) in questa affermazione sono usate per brevità. Non rimuovere nessuna istruzione nella policy di accesso esistente. Non immettere queste ellissi nella policy.

JSON

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

Wrapping

In questo tutorial, creavi AWS CodeBuild un set di file di classe Java in un file JAR. Quindi hai visualizzato i risultati della compilazione.

Ora puoi provare a utilizzarlo CodeBuild nei tuoi scenari. Segui le istruzioni in [Pianifica una compilazione](#). Se non ti senti ancora pronto, puoi provare a compilare alcuni degli esempi. Per ulteriori informazioni, consulta [Usa esempi basati su casi per CodeBuild](#).

Usa esempi basati su casi per CodeBuild

È possibile utilizzare questi esempi basati su casi d'uso per sperimentare: AWS CodeBuild

[Esempi interservizi](#)

Un elenco di esempi interservizi con cui sperimentare. AWS CodeBuild

[Creazione di esempi di badge](#)

Mostra come configurare i badge CodeBuild di costruzione.

[Esempio di rapporto di prova](#)

Utilizza il AWS CLI per creare, eseguire e visualizzare i risultati di un rapporto di test.

[Esempi Docker per CodeBuild](#)

Mostra come utilizzare immagini Docker personalizzate, pubblicare immagini Docker in un repository in Amazon ECR e utilizzare immagini Docker in un registro privato.

[Output di compilazione ospitato in un bucket S3](#)

Illustra come creare un sito Web statico in un bucket S3 utilizzando artefatti della compilazione non crittografati.

[Esempio di ingressi e uscite multipli](#)

Mostra come utilizzare diverse origini di input e numerosi artefatti di output in un progetto di compilazione.

[Esempi di esecuzione di test paralleli](#)

Mostra come utilizzare il comando `codebuild-tests-run` CLI per dividere ed eseguire test in ambienti di esecuzione parallela.

[Versioni runtime nel file buildspec di esempio](#)

Mostra come specificare i runtime e le relative versioni nel file buildspec.

[Esempio di versione di origine](#)

Mostra come utilizzare una versione specifica del codice sorgente in un progetto di CodeBuild compilazione.

[Esempi di repository di sorgenti di terze parti per CodeBuild](#)

Mostra come creare BitBucket, GitHub Enterprise Server ed GitHub estrarre richieste con webhook utilizzando. CodeBuild

[Imposta i nomi degli artefatti in fase di compilazione utilizzando il controllo delle versioni semantiche](#)

Mostra come usare il versioning semantico per generare un nome di artefatto in fase di creazione.

Esempi trasversali per CodeBuild

È possibile utilizzare questi esempi interservizi per sperimentare con: AWS CodeBuild

[Esempio di Amazon ECR](#)

Utilizza un'immagine Docker in un repository Amazon ECR per utilizzare Apache Maven per produrre un singolo file JAR. Le istruzioni di esempio ti mostreranno come creare e inviare un'immagine Docker ad Amazon ECR, creare un progetto Go, creare il progetto, eseguire il progetto e configurare le autorizzazioni per consentire la connessione CodeBuild ad Amazon ECR.

[Esempio di Amazon EFS](#)

Mostra come configurare un file buildspec in modo che un CodeBuild progetto venga montato e costruito su un file system Amazon EFS. Le istruzioni di esempio ti mostreranno come creare un Amazon VPC, creare un file system in Amazon VPC, creare e creare un progetto che utilizza Amazon VPC e quindi rivedere il file di progetto e le variabili generati.

[AWS CodePipeline campioni](#)

Mostra come utilizzare per AWS CodePipeline creare una build con build in batch, più fonti di input e più artefatti di output. In questa sezione sono inclusi file JSON di esempio che mostrano strutture di pipeline che creano build in batch con artefatti separati e artefatti combinati. Viene fornito un esempio JSON aggiuntivo che mostra la struttura della pipeline con più sorgenti di input e più artefatti di output.

[AWS Config campione](#)

Mostra come eseguire la configurazione. AWS Config Elenca CodeBuild le risorse monitorate e descrive come cercare i CodeBuild progetti. AWS Config Le istruzioni di esempio mostreranno i prerequisiti per l'integrazione AWS Config, i passaggi per la configurazione AWS Config e i passaggi in cui cercare CodeBuild progetti e dati. AWS Config

[Esempio di notifiche di compilazione](#)

Utilizza Apache Maven per produrre un singolo file JAR. Invia una notifica di build agli abbonati di un argomento di Amazon SNS. Le istruzioni di esempio mostrano come configurare le autorizzazioni in modo da CodeBuild poter comunicare con Amazon SNS CloudWatch e come creare e CodeBuild identificare argomenti in Amazon SNS, come iscrivere i destinatari all'argomento e come impostare le regole in CloudWatch

Esempio di Amazon ECR per CodeBuild

Questo esempio utilizza un'immagine Docker in un repository di immagini Amazon Elastic Container Registry (Amazon ECR) per creare un progetto Go di esempio.

Important

L'esecuzione di questo esempio potrebbe comportare addebiti sul tuo account. AWS Questi includono possibili addebiti per AWS CodeBuild e per AWS risorse e azioni relative ad Amazon S3 AWS KMS, CloudWatch Logs e Amazon ECR. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i prezzi di Amazon S3, i prezzi, i AWS Key Management Service prezzi di Amazon e CloudWatch i prezzi di Amazon Elastic Container Registry](#).

Argomenti

- [Esegui l'esempio di Amazon ECR](#)

Esegui l'esempio di Amazon ECR

Utilizza le seguenti istruzioni per eseguire l'esempio di Amazon ECR per CodeBuild.

Per eseguire questo esempio

1. Per creare e inviare l'immagine Docker al tuo repository di immagini in Amazon ECR, completa i passaggi nella [Esegui l'esempio «Pubblica immagine Docker su Amazon ECR»](#) sezione del. [Esempio «Pubblica immagine Docker su Amazon ECR»](#)
2. Creazione di un progetto Go:

- a. Crea i file come descritto nelle [File di un progetto Go](#) sezioni [Struttura di un progetto Go](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository AWS CodeCommit, GitHub o Bitbucket.

⚠ Important

Non caricare (*root directory name*), ma solo i file all'interno di (*root directory name*)

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere (*root directory name*) al file ZIP, ma solo i file all'interno di (*root directory name*)

- b. Crea un progetto di compilazione, esegui la build e visualizza le relative informazioni sulla build.

Se usi il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del `create-project` comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

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

```
}
```

- c. Per ottenere l'artefatto di output della compilazione, aprire il bucket di output S3.
 - d. Scaricare il file *GoOutputArtifact.zip* sul computer locale o sull'istanza, quindi estrarre il contenuto del file. Tra i contenuti estratti, selezionare il file `hello`.
3. Se una delle seguenti condizioni è vera, devi aggiungere le autorizzazioni al tuo repository di immagini in Amazon ECR in modo che AWS CodeBuild possa inserire la sua immagine Docker nell'ambiente di compilazione.
- Il tuo progetto utilizza CodeBuild le credenziali per estrarre immagini Amazon ECR. Questo è indicato dal valore `CODEBUILD` nell'attributo `imagePullCredentialsType` di `ProjectEnvironment`.
 - Il tuo progetto utilizza un'immagine Amazon ECR per più account. In questo caso, il progetto deve utilizzare il suo ruolo di servizio per estrarre immagini Amazon ECR. Per abilitare questo comportamento, impostare l'attributo `imagePullCredentialsType` di `ProjectEnvironment` su `SERVICE_ROLE`.
1. Apri la console Amazon ECR all'indirizzo <https://console.aws.amazon.com/ecr/>.
 2. Nella lista dei nomi dei repository, selezionare il nome del repository creato o selezionato.
 3. Dal riquadro di navigazione scegliere Permissions (Autorizzazioni), Edit (Modifica), quindi scegliere Add statement (Aggiungi istruzione).
 4. In Nome istruzione, immettere un identificatore (ad esempio, **CodeBuildAccess**).
 5. Per Effect (Effetto), lasciare selezionato Allow (Abilita). Questo indica che vuoi consentire l'accesso a un altro account AWS .
 6. Per Principal (Principale), eseguire una delle seguenti operazioni:
 - Se il tuo progetto utilizza CodeBuild le credenziali per estrarre un'immagine Amazon ECR, in Service principal, inserisci. **codebuild.amazonaws.com**
 - Se il tuo progetto utilizza un'immagine Amazon ECR per più account, per AWS account IDs inserisci IDs gli AWS account a cui desideri consentire l'accesso.
 7. Saltare la lista All IAM entities (Tutte le entità IAM).
 8. Per Azione, seleziona le azioni di sola estrazione: `ecr:GetDownloadUrlForLayer`, `ecr:` ed `ecr:BatchGetImage` `BatchCheckLayerAvailability`
 9. Per Condizioni, aggiungi quanto segue:

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

10.Scegli Save (Salva).

Questa policy viene visualizzata in Autorizzazioni. L'entità principal è ciò che è stato immesso per Principal (Entità principal) al passaggio 3 di questa procedura:

- Se il tuo progetto utilizza CodeBuild credenziali per estrarre un'immagine Amazon ECR, "codebuild.amazonaws.com" viene visualizzato in Service principals.
- Se il tuo progetto utilizza un'immagine Amazon ECR per più account, l'ID dell' AWS account a cui desideri consentire l'accesso viene visualizzato in AWS Account. IDs

La seguente policy di esempio utilizza sia CodeBuild le credenziali che un'immagine Amazon ECR per più account.

JSON

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

```

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

```

- Se i tuoi progetti utilizzano CodeBuild credenziali e desideri che abbiano accesso aperto all'archivio Amazon ECR, puoi omettere Condition le chiavi e aggiungere la seguente politica di esempio. CodeBuild

JSON

```

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

```

```

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

```

4. Crea un progetto di compilazione, esegui la build e visualizza le informazioni sulla build.

Se usi il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del create-project comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

```

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

```

5. Per ottenere l'artefatto di output della compilazione, aprire il bucket di output S3.
6. Scaricare il file *GoOutputArtifact*.zip sul computer locale o sull'istanza, quindi estrarre il contenuto del file *GoOutputArtifact*.zip. Tra i contenuti estratti, selezionare il file hello.

Struttura di un progetto Go

Questo esempio assume la seguente struttura delle directory.

(root directory name)

```
### buildspec.yml
```

```
### hello.go
```

File di un progetto Go

Questo esempio utilizza i seguenti file.

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

```
version: 0.2

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

hello.go (in *(root directory name)*)

```
package main
import "fmt"

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

```
fmt.Println(true || false)
fmt.Println(!true)
}
```

Esempio di Amazon Elastic File System per AWS CodeBuild

Potresti voler creare le tue AWS CodeBuild build su Amazon Elastic File System, un servizio di file scalabile e condiviso per istanze Amazon EC2. La capacità di storage di Amazon EFS è elastica, quindi aumenta o si riduce man mano che i file vengono aggiunti e rimossi. Ha una semplice interfaccia di servizi Web che ti permette di creare e configurare i file system. Gestisce anche tutta l'infrastruttura di storage dei file, quindi non devi preoccuparti di distribuire, applicare patch o mantenere le configurazioni dei file system. Per ulteriori informazioni, consulta [Cos'è Amazon Elastic File System?](#) nella Amazon Elastic File System User Guide.

Questo esempio mostra come configurare un CodeBuild progetto in modo che monti e poi crei un'applicazione Java su un file system Amazon EFS. Prima di iniziare, è necessario disporre di un'applicazione Java pronta per la creazione che venga caricata in un bucket di input S3 o in un AWS CodeCommit repository GitHub Enterprise Server o Bitbucket. GitHub

I dati in transito per il file system sono crittografati. Per crittografare i dati in transito utilizzando un'immagine diversa, consulta [Crittografia dei dati in transito](#).

Argomenti

- [Utilizzo AWS CodeBuild con Amazon Elastic File System](#)
- [Risolvi i problemi di integrazione con Amazon EFS](#)

Utilizzo AWS CodeBuild con Amazon Elastic File System

L'esempio copre i quattro passaggi di alto livello necessari per utilizzare Amazon EFS con AWS CodeBuild. Questi sono:

1. Crea un cloud privato virtuale (VPC) nel tuo AWS account.
2. Creare un file system che utilizza questo VPC.
3. Crea e costruisci un CodeBuild progetto che utilizza il VPC. Il CodeBuild progetto utilizza quanto segue per identificare il file system:
 - Un identificatore univoco del file system. Si sceglie l'identificatore quando si specifica il file system nel progetto di compilazione.

- L'ID del file system. L'ID viene visualizzato quando visualizzi il file system nella console Amazon EFS.
 - Un punto di montaggio. Questa è una directory nel contenitore Docker che monta il file system.
 - Opzioni di montaggio. Includono dettagli su come montare il file system.
4. Esamina il progetto di compilazione per assicurarti che siano stati generati i file e le variabili di progetto corretti.

Note

Un file system creato in Amazon EFS è supportato solo su piattaforme Linux.

Argomenti

- [Fase 1: Creare un VPC utilizzando CloudFormation](#)
- [Fase 2: crea un file system Amazon Elastic File System con il tuo VPC](#)
- [Fase 3: creare un CodeBuild progetto da utilizzare con Amazon EFS](#)
- [Passaggio 4: rivedi il progetto di compilazione](#)

Fase 1: Creare un VPC utilizzando CloudFormation

Crea il tuo VPC con un CloudFormation modello.

1. Segui le istruzioni [CloudFormation Modello VPC](#) per l'uso CloudFormation per creare un VPC.

Note

Il VPC creato da questo CloudFormation modello ha due sottoreti private e due sottoreti pubbliche. È necessario utilizzare solo sottoreti private quando si utilizza AWS CodeBuild per montare il file system creato in Amazon EFS. Se si utilizza una delle sottoreti pubbliche, la build ha esito negativo.

2. Accedi Console di gestione AWS e apri la console Amazon VPC all'indirizzo. <https://console.aws.amazon.com/vpc/>
3. Scegli il VPC con cui hai creato. CloudFormation

4. Prendere nota del nome del VPC e dell'ID presenti nella scheda Description (Descrizione). Questo ID sarà necessario al momento della creazione del progetto AWS CodeBuild più avanti in questo esempio.

Fase 2: crea un file system Amazon Elastic File System con il tuo VPC

Crea un semplice file system Amazon EFS per questo esempio utilizzando il VPC creato in precedenza.

1. Accedi Console di gestione AWS e apri la console Amazon EFS all'indirizzo <https://console.aws.amazon.com/efs/>.
2. Scegliere Create file system (Crea file system).
3. In VPC scegliere il nome VPC annotato in precedenza in questo esempio.
4. Lasciare le zone di disponibilità associate alle sottoreti selezionate.
5. Selezionare Next Step (Fase successiva).
6. In Aggiungi tag, per la chiave Name predefinita, in Value, inserisci il nome del tuo file system Amazon EFS.
7. Mantenere Bursting e General Purpose (Utilizzo generico) selezionati come performance predefinita e modalità di throughput, quindi scegliere Next Step (Fase successiva).
8. Per Configure client access (Configura accesso client), scegliere Next Step (Passaggio successivo).
9. Scegliere Create File System (Crea file system).
10. (Facoltativo) Ti consigliamo di aggiungere una policy al tuo file system Amazon EFS che applichi la crittografia dei dati in transito. Nella console Amazon EFS, scegli Politica del file system, scegli Modifica, seleziona la casella Applica la crittografia in transito per tutti i client, quindi scegli Salva.

Fase 3: creare un CodeBuild progetto da utilizzare con Amazon EFS

Crea un AWS CodeBuild progetto che utilizzi il VPC creato in precedenza in questo esempio. Quando la build viene eseguita, monta il file system Amazon EFS creato in precedenza. Successivamente, memorizza il file.jar creato dall'applicazione Java nella directory del punto di montaggio del file system.

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Nel riquadro di navigazione scegliere Build projects (Progetti di compilazione), quindi Create build project (Crea progetto di compilazione).
3. In Project name (Nome progetto) immettere un nome per il progetto.
4. In Source provider (Provider di origine) scegliere il repository contenente l'applicazione Java che si desidera compilare.
5. Inserisci le informazioni, come l'URL del repository, da utilizzare per localizzare l' CodeBuild applicazione. Le opzioni variano a seconda del provider di origine. Per ulteriori informazioni, consulta [Choose source provider](#).
6. In Environment image (Immagine ambiente) scegliere Managed image (Immagine gestita).
7. In Operating system (Sistema operativo), scegliere Amazon Linux 2.
8. In Runtime(s) (Runtime), seleziona Standard.
9. Da Image, scegliete aws/codebuild/amazonlinux-x86_64-standard:4.0.
10. Da Environment type (Tipo ambiente), seleziona Linux.
11. In Service role (Ruolo del servizio) scegliere New service role (Nuovo ruolo del servizio). In Nome ruolo, inserisci un nome per il ruolo creato per te. CodeBuild
12. Espandere Additional configuration (Configurazione aggiuntiva).
13. Seleziona Enable this flag if you want to build Docker images or want your builds to get elevated privileges (Abilita questo flag se desideri creare immagini Docker o se desideri che le build ottengano privilegi elevati).

 Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs e](#) abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

14. In VPC scegliere l'ID VPC.
15. In Subnets (Sottoreti) scegliere una o più sottoreti private associate al VPC. È necessario utilizzare sottoreti private in una build che monta un file system Amazon EFS. Se si utilizza una sottorete pubblica, la build ha esito negativo.
16. Da Security Group (Gruppo di sicurezza), scegliere il gruppo di sicurezza predefinito.
17. In File Systems (File system), immettere le seguenti informazioni:

- Per Identifier, immettere un identificatore univoco del file system. Deve contenere meno di 129 caratteri e solo caratteri alfanumerici e caratteri di sottolineatura. CodeBuild utilizza questo identificatore per creare una variabile di ambiente che identifichi il file system elastico. Il formato della variabile di ambiente è `CODEBUILD_<file_system_identifier>` in lettere maiuscole. Ad esempio, se si immette `my_efs`, la variabile di ambiente è `CODEBUILD_MY_EFS`.
- Per ID, scegliere l'ID del file system.
- (Facoltativo) Inserisci una directory nel file system. CodeBuild monta questa directory. Se si lascia vuoto il percorso della directory, CodeBuild monta l'intero file system. Il percorso del file è relativo rispetto alla root del file system.
- Per Mount point, inserisci il percorso assoluto della directory nel contenitore di build in cui è montato il file system. Se questa directory non esiste, la CodeBuild crea durante la compilazione.
- (Facoltativo) Immettere le opzioni di montaggio. Se lasci vuote le opzioni di montaggio, CodeBuild utilizza le opzioni di montaggio predefinite:

```
nfsvers=4.1
rsiz=1048576
wsiz=1048576
hard
timeo=600
retrans=2
```

Per ulteriori informazioni, consulta le [opzioni di montaggio NFS consigliate](#) nella Amazon Elastic File System User Guide.

18. Per Build specification (Specifiche di compilazione) scegliere Insert build commands (Inserisci comandi di compilazione) e selezionare Switch to editor (Passa a editor).
19. Inserisci i seguenti comandi build spec nell'editor. Sostituire `<file_system_identifier>` con l'identificatore inserito nel passaggio 17. Utilizzare lettere maiuscole (ad esempio, `CODEBUILD_MY_EFS`).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
```

```
build:
  commands:
    - mvn compile -Dgpg.skip=true -Dmaven.repo.local=
      $CODEBUILD_<file_system_identifier>
```

20. Utilizzare i valori di default per tutte le altre impostazioni, quindi scegliere Create build project (Crea progetto di compilazione). Al termine della compilazione viene visualizzata la pagina della console per il progetto.
21. Selezionare Start build (Avvia compilazione).

Passaggio 4: rivedi il progetto di compilazione

Dopo aver creato il AWS CodeBuild progetto:

- Hai un file.jar creato dalla tua applicazione Java che è integrato nel tuo file system Amazon EFS nella tua directory dei punti di montaggio.
- Una variabile di ambiente che identifichi il file system viene creata utilizzando l'identificatore del file system immesso al momento della creazione del progetto.

Per ulteriori informazioni, consulta [Mounting file system](#) nella Amazon Elastic File System User Guide.

Risolvi i problemi di integrazione con Amazon EFS

Di seguito sono riportati gli errori che potresti riscontrare durante la configurazione di Amazon EFS con CodeBuild.

Argomenti

- [CLIENT_ERROR: montaggio di '127.0.0.1: /' non riuscito. Autorizzazione negata](#)
- [CLIENT_ERROR: montaggio di '127.0.0.1: /' non riuscito. Reimpostazione della connessione da parte di un peer](#)
- [VPC_CLIENT_ERROR: errore EC2 imprevisto: UnauthorizedOperation](#)

CLIENT_ERROR: montaggio di '127.0.0.1: /' non riuscito. Autorizzazione negata

L'autorizzazione IAM non è supportata per il montaggio di Amazon EFS con CodeBuild. Se utilizzi una policy del file system Amazon EFS personalizzata, dovrai concedere l'accesso in lettura e scrittura a tutti i principali IAM. Esempio:

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

CLIENT_ERROR: montaggio di '127.0.0.1: /' non riuscito. Reimpostazione della connessione da parte di un peer

Esistono due possibili cause di questo errore:

- La sottorete CodeBuild VPC si trova in una zona di disponibilità diversa rispetto alla destinazione di montaggio di Amazon EFS. Puoi risolvere questo problema aggiungendo una sottorete VPC nella stessa zona di disponibilità del target di montaggio Amazon EFS.
- Il gruppo di sicurezza non dispone delle autorizzazioni per comunicare con Amazon EFS. Puoi risolvere questo problema aggiungendo una regola in entrata per consentire tutto il traffico proveniente dal VPC (aggiungi il blocco CIDR primario per il tuo VPC) o dal gruppo di sicurezza stesso.

VPC_CLIENT_ERROR: errore EC2 imprevisto: UnauthorizedOperation

Questo errore si verifica quando tutte le sottoreti nella configurazione VPC per il CodeBuild progetto sono sottoreti pubbliche. È necessario disporre di almeno una sottorete privata nel VPC per garantire la connettività di rete.

AWS CodePipeline campioni per CodeBuild

Questa sezione descrive esempi di integrazioni tra CodePipeline e CodeBuild.

Project N.E.M.O.	Description
Esempi di CodePipeline/CodeBuild integrazioni e build in batch	Questi esempi mostrano come utilizzare per AWS CodePipeline creare un progetto di compilazione che utilizza build in batch.

Project N.E.M.O.	Description
Esempio di CodePipeline/CodeBuild integrazione con più sorgenti di input e artefatti di output	Questo esempio dimostra come utilizzarlo per AWS CodePipeline creare un progetto di compilazione che utilizza più fonti di input per creare più artefatti di output.

Esempi di CodePipeline/CodeBuild integrazioni e build in batch

AWS CodeBuild supporta le build in batch. Gli esempi seguenti mostrano come utilizzare per AWS CodePipeline creare un progetto di compilazione che utilizza build in batch.

È possibile utilizzare un file in formato JSON che definisce la struttura della pipeline e quindi utilizzarlo con il per creare la pipeline. AWS CLI Per ulteriori informazioni, consulta il riferimento alla struttura della [AWS CodePipeline pipeline](#) nella Guida per l'utente.AWS CodePipeline

Creazione in batch con singoli artefatti

Utilizza il seguente file JSON come esempio di struttura di pipeline che crea una build in batch con artefatti separati. Per abilitare la compilazione in batch CodePipeline, impostate il BatchEnabled parametro dell'oggetto su. configuration true

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

```

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

```

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

Di seguito è riportato un esempio di file CodeBuild buildspec che funzionerà con questa configurazione della pipeline.


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

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

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

I nomi degli artefatti di output specificati nel file JSON della pipeline devono corrispondere all'identificatore delle build e degli artefatti definiti nel file buildspec. La sintassi è per gli artefatti primari e _ per gli artefatti *buildIdentifier* secondari. *buildIdentifier artifactIdentifier*

Ad esempio, per il nome dell'artefatto di output build1, CodeBuild caricherà l'artefatto principale nella posizione di. build1 build1 Per il nome di output build1_artifact1, CodeBuild caricherà l'artefatto secondario di nella posizione artifact1 build1 di, e così via. build1_artifact1 Se viene specificata una sola posizione di output, il nome deve essere *buildIdentifier* solo.

Dopo aver creato il file JSON, puoi creare la tua pipeline. Utilizzate il AWS CLI per eseguire il comando create-pipeline e passate il file al parametro. --cli-input-json Per ulteriori informazioni, consulta [Create a pipeline \(CLI\)](#) nella Guida per AWS CodePipeline l'utente.

Creazione in batch con artefatti combinati

Utilizzate il seguente file JSON come esempio di struttura di pipeline che crea una build in batch con artefatti combinati. Per abilitare la compilazione in batch CodePipeline, impostate il `BatchEnabled` parametro dell'oggetto su. `configuration true` Per combinare gli elementi di costruzione nella stessa posizione, impostate il `CombineArtifacts` parametro dell'`configuration` oggetto su. `true`

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "<my-input-bucket-name>",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          },
          {
            "inputArtifacts": [],
            "name": "Source2",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            }
          }
        ]
      }
    ]
  }
}
```

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

```

```
        "runOrder": 1
      }
    ]
  },
  "artifactStore": {
    "type": "S3",
    "location": "<AWS-CodePipeline-internal-bucket-name>"
  },
  "name": "my-pipeline-name",
  "version": 1
}
```

Di seguito è riportato un esempio di file CodeBuild buildspec che funzionerà con questa configurazione della pipeline.

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

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

artifacts:
  files:
    - output_file
```

Se gli artefatti combinati sono abilitati per la compilazione in batch, è consentito un solo output. CodeBuild combinerà gli artefatti principali di tutte le build in un unico file ZIP.

Dopo aver creato il file JSON, puoi creare la tua pipeline. Utilizzate il AWS CLI per eseguire il comando `create-pipeline` e passare il file al parametro. `--cli-input-json` Per ulteriori informazioni, consulta [Create a pipeline \(CLI\)](#) nella Guida per AWS CodePipeline l'utente.

Esempio di CodePipeline/CodeBuild integrazione con più sorgenti di input e artefatti di output

Un AWS CodeBuild progetto può richiedere più di una fonte di input. Può anche creare più di un artefatto di output. Questo esempio dimostra come utilizzarlo per AWS CodePipeline creare un progetto di compilazione che utilizzi più fonti di input per creare più artefatti di output. Per ulteriori informazioni, consulta [Diverse origini di input ed esempi di artefatti di output](#).

È possibile utilizzare un file in formato JSON che definisce la struttura della pipeline e quindi utilizzarlo con il per creare la pipeline. AWS CLI Utilizzare il seguente file JSON come esempio di una struttura di pipeline che crea una compilazione con più di un'origine di input e più di un artefatto di output. Più avanti in questo esempio viene illustrato come questo file specifica i numerosi input e output. Per ulteriori informazioni, consulta il riferimento alla struttura della [CodePipeline pipeline](#) nella Guida per l'utente.AWS CodePipeline

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

```

```
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "my-other-input-bucket-name",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "AWS CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "artifact1"
        }
      ]
    }
  ]
}
```

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

In questo file JSON:

- Una delle origini di input deve essere un percorso PrimarySource. Questa fonte è la directory in cui CodeBuild cerca ed esegue il file buildspec. La parola chiave PrimarySource viene utilizzata per specificare la fonte primaria nella configuration sezione dello CodeBuild stage del file JSON.
- Ogni origine di input è installata nella propria directory. Questa directory viene archiviata nella variabile di ambiente integrata \$CODEBUILD_SRC_DIR per l'origine principale e in \$CODEBUILD_SRC_DIR_yourInputArtifactName per tutte le altre origini. Per la pipeline di questo esempio, le due directory delle origini di input sono \$CODEBUILD_SRC_DIR e \$CODEBUILD_SRC_DIR_source2. Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).
- I nomi degli artefatti di output specificati nel file JSON della pipeline devono corrispondere ai nomi degli artefatti secondari definiti nel file buildspec. Questa pipeline utilizza il seguente file buildspec. Per ulteriori informazioni, consulta [Sintassi buildspec](#).

```
version: 0.2

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

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

Dopo aver creato il file JSON, puoi creare la tua pipeline. Utilizzate il AWS CLI per eseguire il comando `create-pipeline` e passate il file al parametro. `--cli-input-json` Per ulteriori informazioni, consulta [Create a pipeline \(CLI\)](#) nella Guida per AWS CodePipeline l'utente.

AWS Config campionare con CodeBuild

AWS Config fornisce un inventario delle AWS risorse e una cronologia delle modifiche alla configurazione di tali risorse. AWS Config ora supporta AWS CodeBuild come AWS risorsa, il che significa che il servizio può tracciare i tuoi CodeBuild progetti. Per ulteriori informazioni su AWS Config, consulta [What is AWS Config?](#) nella Guida per gli AWS Config sviluppatori.

Puoi visualizzare le seguenti informazioni sulle CodeBuild risorse nella pagina Inventario delle risorse della AWS Config console:

- Una cronologia delle modifiche alla CodeBuild configurazione.
- Dettagli di configurazione per ogni CodeBuild progetto.
- Rapporti con altre AWS risorse.

- Un elenco di modifiche ai tuoi CodeBuild progetti.

Argomenti

- [Usare CodeBuild con AWS Config](#)
- [Passaggio 3: Visualizza AWS CodeBuild i dati nella console AWS Config](#)

Usare CodeBuild con AWS Config

Le procedure riportate in questo argomento mostrano come impostare AWS Config e cercare CodeBuild progetti.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Configurazione AWS Config](#)
- [Passaggio 2: cerca i AWS CodeBuild progetti](#)

Prerequisiti

Crea il tuo AWS CodeBuild progetto. Per istruzioni, consulta [Creazione di un progetto di compilazione](#).

Fase 1: Configurazione AWS Config

- [Impostazione di AWS Config \(console\)](#)
- [Impostazione di AWS Config \(AWS CLI\)](#)

Note

Dopo aver completato la configurazione, potrebbero essere necessari fino a 10 minuti prima di poter visualizzare AWS CodeBuild i progetti nella AWS Config console.

Passaggio 2: cerca i AWS CodeBuild progetti

1. Accedi alla console di AWS gestione e apri la AWS Config console all'indirizzo <https://console.aws.amazon.com/config>.

2. Nella pagina Inventario delle risorse, seleziona AWS CodeBuild Progetto in Tipo di risorsa. Scorri verso il basso e seleziona la casella di controllo CodeBuild del progetto.
3. Scegliere Look up (Cerca).
4. Dopo aver aggiunto l'elenco dei CodeBuild progetti, scegli il link del nome CodeBuild del progetto nella colonna Config timeline.

Passaggio 3: Visualizza AWS CodeBuild i dati nella console AWS Config

Quando cerchi le risorse nella pagina Inventario delle risorse, puoi scegliere la AWS Config sequenza temporale per visualizzare i dettagli CodeBuild del tuo progetto. La pagina dei dettagli per una risorsa fornisce informazioni sulla configurazione, sulle relazioni e sul numero di modifiche apportate a tale risorsa.

I blocchi nella parte superiore della pagina sono complessivamente denominati timeline. La timeline mostra la data e l'ora in cui è stata effettuata la registrazione.

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di configurazione nella AWS Config console](#) nella Guida per gli AWS Config sviluppatori.

Crea un esempio di notifiche per CodeBuild

Amazon CloudWatch Events dispone di un supporto integrato per AWS CodeBuild. CloudWatch Events è un flusso di eventi di sistema che descrive i cambiamenti nelle tue AWS risorse. Con CloudWatch Events, scrivi regole dichiarative per associare gli eventi di interesse alle azioni automatiche da intraprendere. Questo esempio utilizza Amazon CloudWatch Events e Amazon Simple Notification Service (Amazon SNS) per inviare notifiche di build agli abbonati ogni volta che le build hanno successo, falliscono, passano da una fase di compilazione all'altra o qualsiasi combinazione di questi eventi.

Important

L'esecuzione di questo esempio potrebbe comportare addebiti sul tuo account. AWS Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon CloudWatch e Amazon SNS. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i CloudWatch prezzi di Amazon](#) e i prezzi di [Amazon SNS](#).

Argomenti

- [Esegui l'esempio di notifiche di compilazione](#)
- [Informazioni di riferimento sul formato di input delle notifiche di compilazione](#)

Esegui l'esempio di notifiche di compilazione

Utilizza la procedura seguente per eseguire l'esempio di notifiche di compilazione.

Per eseguire questo esempio

1. Se hai già configurato e sottoscritto un argomento in Amazon SNS che desideri utilizzare per questo esempio, vai avanti al passaggio 4. Altrimenti, se utilizzi un utente IAM anziché un account AWS root o un utente amministratore per lavorare con Amazon SNS, aggiungi la seguente dichiarazione (tra **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) all'utente (o al gruppo IAM a cui l'utente è associato). L'utilizzo di un account AWS root non è consigliato. Questa dichiarazione consente di visualizzare, creare, sottoscrivere e testare l'invio di notifiche agli argomenti in Amazon SNS. Le ellissi (. . .) vengono utilizzate per brevità e per aiutare i clienti a individuare il punto in cui aggiungere l'istruzione. Non rimuovere nessuna istruzione e non digitare queste ellissi nella policy esistente.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

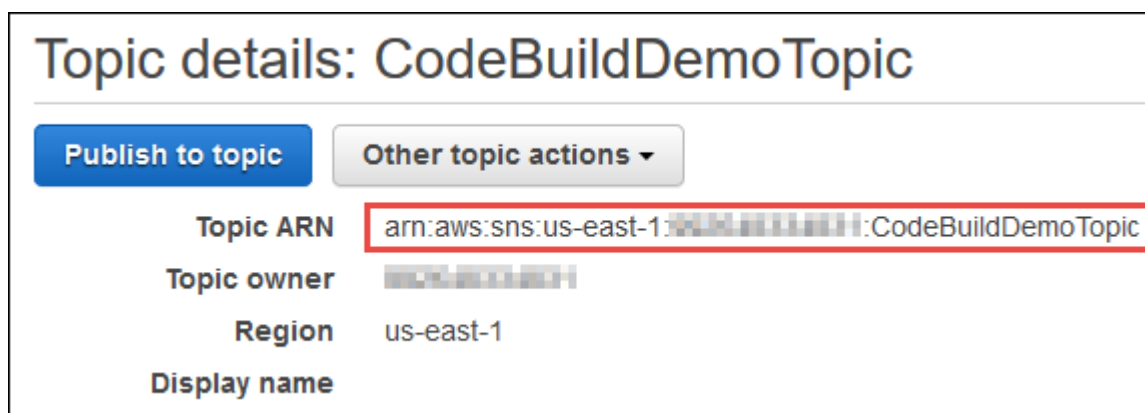
L'entità IAM che modifica questa politica deve disporre dell'autorizzazione in IAM per modificare le politiche.

Per ulteriori informazioni, consulta [Modifica delle politiche gestite dai clienti](#) o la sezione «Modificare o eliminare una politica in linea per un gruppo, un utente o un ruolo» in [Working with inline policies \(console\)](#) nella IAM User Guide.

2. Crea o identifica un argomento in Amazon SNS. AWS CodeBuild utilizza CloudWatch Events per inviare notifiche di build su questo argomento tramite Amazon SNS.

Per creare un argomento:

1. [Apri la console Amazon SNS in https://console.aws.amazon.com/sns](https://console.aws.amazon.com/sns).
2. Scegli Create topic (Crea argomento).
3. In Create new topic (Crea nuovo argomento) per Topic name (Nome argomento), digitare un nome per l'argomento, ad esempio **CodeBuildDemoTopic**. (Se si sceglie un nome diverso, sostituirlo in tutto il campione).
4. Scegli Create topic (Crea argomento).
5. Nella CodeBuildDemoTopic pagina Dettagli argomento:, copia il valore Arn dell'argomento. che servirà per la fase successiva.

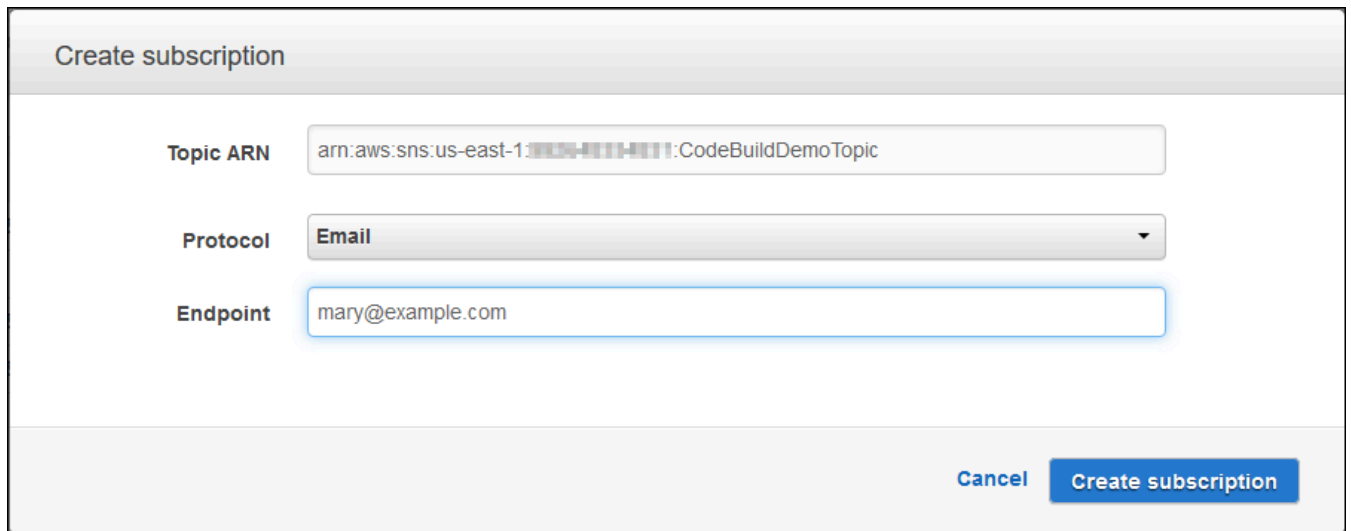


Per ulteriori informazioni, consulta [Creare un argomento](#) nella Amazon SNS Developer Guide.

3. Abbonare uno o più destinatari all'argomento per ricevere notifiche e-mail.

Per sottoscrivere un destinatario a un argomento:

1. Con la console Amazon SNS aperta dal passaggio precedente, nel riquadro di navigazione, scegli Abbonamenti, quindi scegli Crea abbonamento.
2. In Create subscription (Crea abbonamento), per Topic ARN (ARN argomento), copiare e incollare l'ARN dell'argomento dalla fase precedente.
3. Per Protocollo, scegli E-mail.
4. Per Endpoint, immettere l'indirizzo e-mail completo del destinatario.



The screenshot shows the 'Create subscription' form in the Amazon SNS console. The form is titled 'Create subscription' and contains three input fields: 'Topic ARN' with the value 'arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic', 'Protocol' set to 'Email', and 'Endpoint' with the value 'mary@example.com'. At the bottom right of the form are two buttons: 'Cancel' and 'Create subscription'.

5. Selezionare Crea sottoscrizione.
6. Amazon SNS invia un'e-mail di conferma dell'abbonamento al destinatario. Per iniziare a ricevere le notifiche e-mail, il destinatario deve selezionare il link Confirm subscription (Conferma abbonamento) nell'e-mail di conferma dell'abbonamento. Dopo che il destinatario ha fatto clic sul link, se l'iscrizione è avvenuta correttamente, Amazon SNS visualizza un messaggio di conferma nel browser Web del destinatario.

Per ulteriori informazioni, consulta [Abbonarsi a un argomento](#) nella Amazon SNS Developer Guide.

4. Se utilizzi un utente anziché un account AWS root o un utente amministratore per lavorare con CloudWatch Events, aggiungi la seguente dichiarazione (tra **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) all'utente (o al gruppo IAM a cui l'utente è associato). L'utilizzo di un account AWS root non è consigliato. Questa istruzione viene utilizzata per consentire all'utente di lavorare con CloudWatch Events. Le ellissi (. . .) vengono

utilizzate per brevità e per aiutare i clienti a individuare il punto in cui aggiungere l'istruzione. Non rimuovere nessuna istruzione e non digitare queste ellissi nella policy esistente.

JSON

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

Note

L'entità IAM che modifica questa politica deve disporre dell'autorizzazione in IAM per modificare le politiche.

Per ulteriori informazioni, consulta [Modifica delle politiche gestite dai clienti](#) o la sezione «Modificare o eliminare una politica in linea per un gruppo, un utente o un ruolo» in [Working with inline policies \(console\)](#) nella IAM User Guide.

5. Crea una regola in CloudWatch Events. Per fare ciò, apri la CloudWatch console, in <https://console.aws.amazon.com/cloudwatch>.
6. Nel riquadro di navigazione, in Events (Eventi), selezionare Rules (Regolamento), quindi Create rule (Crea regolamento).
7. Nella pagina Step 1: Create rule (Fase 1: Creare regolamento), dovrebbero già essere selezionate le opzioni Event Pattern (Modello evento) e Build event pattern to match events by service (Crea un modello evento da abbinare agli eventi in base al servizio).
8. Per Service Name (Nome del servizio), selezionare CodeBuild. Per Event Type (Tipo di evento), All Events (Tutti gli eventi) dovrebbe già essere selezionato.
9. Il codice seguente deve essere visualizzato nell'anteprima del modello di evento:

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

10. Scegliete **Modifica** e sostituite il codice in **Event Pattern Preview** con uno dei due modelli di regole seguenti.

Il primo modello di regola attiva un evento quando una compilazione viene avviata o completata, per i progetti di compilazione specificati in AWS CodeBuild.

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

Nella regola precedente, eseguire le modifiche al codice seguente laddove necessario.

- Per attivare un evento quando una compilazione viene avviata o completata, lasciare tutti i valori come illustrato nell'array `build-status` oppure rimuovere completamente l'array `build-status`.
- Per attivare un evento solo quando una compilazione viene completata, rimuovere `IN_PROGRESS` dall'array `build-status`.

- Per attivare un evento solo quando una compilazione viene avviata, rimuovere tutti i valori eccetto `IN_PROGRESS` dall'array `build-status`.
- Per attivare gli eventi per tutti i progetti di compilazione, rimuovere completamente l'array `project-name`.
- Per attivare gli eventi solo per singoli progetti di compilazione, specificare il nome di ciascun progetto di compilazione nell'array `project-name`.

Il secondo modello di regola attiva un evento ogni volta che una compilazione si sposta da una fase di compilazione a un'altra, per i progetti di compilazione specificati in AWS CodeBuild.

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



```
}  
}
```

Nella regola precedente, eseguire le modifiche al codice seguente laddove necessario.

- Per attivare un evento per ogni modifica della fase di compilazione (che può inviare fino a nove notifiche per ogni compilazione), lasciare tutti i valori come illustrato nell'array `completed-phase` oppure rimuovere completamente l'array `completed-phase`.
- Per attivare gli eventi solo per le singole modifiche della fase di compilazione, rimuovere il nome di ciascun progetto di compilazione nell'array `completed-phase` per cui non si desidera attivare un evento.
- Per attivare un evento per ogni modifica dello stato della fase di compilazione, lasciare tutti i valori come illustrato nell'array `completed-phase-status` oppure rimuovere completamente l'array `completed-phase-status`.
- Per attivare gli eventi solo per le singole modifiche dello stato della fase di compilazione, rimuovere il nome di ciascuno stato della fase di compilazione nell'array `completed-phase-status` per cui non si desidera attivare un evento.
- Per attivare gli eventi per tutti i progetti di compilazione, rimuovi l'array `project-name`.
- Per attivare gli eventi per i singoli progetti di compilazione, specificare il nome di ciascun progetto di compilazione nell'array `project-name`.

Per ulteriori informazioni sui pattern di eventi, consulta [Event Patterns](#) nella Amazon EventBridge User Guide.

Per ulteriori informazioni sul filtraggio con pattern di eventi, consulta [Content-based Filtering with Event Patterns](#) nella Amazon EventBridge User Guide.

Note

Se si desidera attivare gli eventi sia per le modifiche dello stato della compilazione che per le modifiche della fase di compilazione, è necessario creare due regole separate, una per le modifiche dello stato della compilazione e l'altra per le modifiche della fase di compilazione. Se si tenta di combinare entrambe le regole in una singola regola, la regola combinata potrebbe produrre risultati imprevisti o smettere di funzionare del tutto.

Al termine della sostituzione del codice, selezionare Save (Salva).

11. In Targets (Destinazioni), seleziona Add target (Aggiungi destinazione).
12. Nell'elenco degli obiettivi, selezionare SNS topic (Argomento SNS);
13. Per Argomento, selezionare l'argomento identificato o creato in precedenza.
14. Espandere Configure input (Configura input), quindi selezionare Input Transformer (Trasformatore di input).
15. Nella casella Input Path (Percorso dell'input), immettere uno dei seguenti percorsi di input.

Per una regola con il valore CodeBuild Build State Change per detail-type, immettere quanto segue.

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

Per una regola con il valore CodeBuild Build Phase Change per detail-type, immettere quanto segue.

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

Per ottenere altri tipi di informazioni, consultare [Informazioni di riferimento sul formato di input delle notifiche di compilazione](#).

16. Nella casella Input Template (Modello dell'input), immettere uno dei seguenti modelli di input.

Per una regola con il valore CodeBuild Build State Change per detail-type, immettere quanto segue.

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

Per una regola con il valore CodeBuild Build Phase Change per detail-type, immettere quanto segue.

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

17. Scegli Configura dettagli.
18. Nella pagina Step 2: Configure rule details (Fase 2: Configurare i dettagli della regola), immettere un nome e una descrizione facoltativa. Per State (Stato), lasciare l'opzione Enabled (Abilitato) selezionata.
19. Scegli Crea regola.
20. Crea progetti di compilazione, esegui le build e visualizza le informazioni sulle build.
21. Conferma che CodeBuild ora le notifiche di build sono state inviate correttamente. Ad esempio, verificare se le e-mail di notifica di compilazione si trovano nella casella di posta.

Per modificare il comportamento di una regola, nella CloudWatch console, scegli la regola che desideri modificare, scegli Azioni e quindi scegli Modifica. Apportare le modifiche alla regola, scegliere Configure details (Configura dettagli) e quindi selezionare Update rule (Aggiorna regola).

Per smettere di usare una regola per inviare notifiche di build, nella CloudWatch console, scegli la regola che vuoi smettere di usare, scegli Azioni e quindi scegli Disabilita.

Per eliminare completamente una regola, nella CloudWatch console, scegli la regola che desideri eliminare, scegli Azioni e quindi scegli Elimina.

Informazioni di riferimento sul formato di input delle notifiche di compilazione

CloudWatch invia notifiche in formato JSON.

Le notifiche di modifica dello stato di compilazione utilizzano il seguente formato:

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ]
}
```

```

],
"detail":{
  "build-status": "SUCCEEDED",
  "project-name": "my-sample-project",
  "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
  "additional-information": {
    "artifact": {
      "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
      "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
      "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
    },
    "environment": {
      "image": "aws/codebuild/standard:5.0",
      "privileged-mode": false,
      "compute-type": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environment-variables": []
    },
    "timeout-in-minutes": 60,
    "build-complete": true,
    "initiator": "MyCodeBuildDemoUser",
    "build-start-time": "Sep 1, 2017 4:12:29 PM",
    "source": {
      "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
      "type": "S3"
    },
    "logs": {
      "group-name": "/aws/codebuild/my-sample-project",
      "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
      "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
    },
    "phases": [
      {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:12:29 PM",
        "end-time": "Sep 1, 2017 4:12:29 PM",
        "duration-in-seconds": 0,
        "phase-type": "SUBMITTED",
        "phase-status": "SUCCEEDED"
      }
    ]
  }
}

```

```
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:12:29 PM",
  "end-time": "Sep 1, 2017 4:13:05 PM",
  "duration-in-seconds": 36,
  "phase-type": "PROVISIONING",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:05 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 4,
  "phase-type": "DOWNLOAD_SOURCE",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
```

```

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

```

Le notifiche di modifica della fase di compilazione utilizzano il seguente formato:

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",

```

```

"resources":[
  "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
],
"detail":{
  "completed-phase": "COMPLETED",
  "project-name": "my-sample-project",
  "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
  "completed-phase-context": "[]",
  "additional-information": {
    "artifact": {
      "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
      "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
      "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
    },
    "environment": {
      "image": "aws/codebuild/standard:5.0",
      "privileged-mode": false,
      "compute-type": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environment-variables": []
    },
    "timeout-in-minutes": 60,
    "build-complete": true,
    "initiator": "MyCodeBuildDemoUser",
    "build-start-time": "Sep 1, 2017 4:12:29 PM",
    "source": {
      "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
      "type": "S3"
    },
    "logs": {
      "group-name": "/aws/codebuild/my-sample-project",
      "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
      "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
    },
    "phases": [
      {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:12:29 PM",

```

```
    "end-time": "Sep 1, 2017 4:12:29 PM",
    "duration-in-seconds": 0,
    "phase-type": "SUBMITTED",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:12:29 PM",
    "end-time": "Sep 1, 2017 4:13:05 PM",
    "duration-in-seconds": 36,
    "phase-type": "PROVISIONING",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:05 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 4,
    "phase-type": "DOWNLOAD_SOURCE",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "INSTALL",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:13:10 PM",
    "duration-in-seconds": 0,
    "phase-type": "PRE_BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:13:10 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 70,
    "phase-type": "BUILD",
    "phase-status": "SUCCEEDED"
  }
```



```

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

```

Crea un esempio di badge con CodeBuild

AWS CodeBuild ora supporta l'uso dei build badge, che forniscono un'immagine incorporabile e generata dinamicamente (badge) che mostra lo stato dell'ultima build di un progetto. Questa

immagine è accessibile tramite un URL disponibile pubblicamente generato per il tuo progetto. CodeBuild Ciò consente a chiunque di visualizzare lo stato di un CodeBuild progetto. I badge di compilazione non contengono alcuna informazione di sicurezza, quindi non richiedono l'autenticazione.

Argomenti

- [Crea un progetto di compilazione con i badge di costruzione](#)
- [Accedi ai badge di costruzione AWS CodeBuild](#)
- [Pubblica badge di costruzione CodeBuild](#)
- [CodeBuild stati dei badge](#)

Crea un progetto di compilazione con i badge di costruzione

Utilizza una delle seguenti procedure per creare un progetto di compilazione con i badge di costruzione abilitati. Puoi usare AWS CLI o il Console di gestione AWS

Per creare un progetto di compilazione con i badge di compilazione abilitati (AWS CLI)

- Per informazioni sulla creazione del progetto di compilazione, consultare [Creazione di un progetto di compilazione \(AWS CLI\)](#). Per includere i badge di costruzione nel AWS CodeBuild progetto, è necessario specificare *badgeEnabled* con un valore di `true`

Per creare un progetto di compilazione con i badge di compilazione abilitati (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
4. In Source (Origine), per Source provider (Provider di origine), selezionare il tipo di provider di codice sorgente ed eseguire una delle operazioni seguenti:

 Note

CodeBuild non supporta i build badge con il provider di sorgenti Amazon S3. Poiché AWS CodePipeline utilizza Amazon S3 per il trasferimento di artefatti, i badge di costruzione non sono supportati per i progetti di compilazione che fanno parte di una pipeline creata in CodePipeline.

- Se si seleziona CodeCommit in seguito per Repository (Archivio), selezionare il nome dell'archivio. Selezionare Enable build badge (Abilita badge di compilazione) per rendere visibile e integrabile lo stato della compilazione del progetto.
- Se lo desideri GitHub, segui le istruzioni per connetterti (o riconnetterti) con GitHub. Nella pagina GitHub Autorizza l'applicazione, per l'accesso all'organizzazione, scegli Richiedi accesso accanto a ciascun repository AWS CodeBuild a cui desideri poter accedere. Dopo aver selezionato Authorize application (Autorizza applicazione), nella console AWS CodeBuild, per Repository (Archivio), selezionare il nome dell'archivio che contiene il codice sorgente. Selezionare Enable build badge (Abilita badge di compilazione) per rendere visibile e integrabile lo stato della compilazione del progetto.
- Se si è selezionato Bitbucket, seguire le istruzioni per connettersi (o riconnettersi) a Bitbucket. Nella pagina Bitbucket Confirm access to your account (Conferma l'accesso all'account), per Organization access (Accesso organizzazione), selezionare Grant access (Concedi accesso). Dopo aver scelto Concedi l'accesso, torna nella AWS CodeBuild console, per Repository, scegli il nome del repository che contiene il codice sorgente. Selezionare Enable build badge (Abilita badge di compilazione) per rendere visibile e integrabile lo stato della compilazione del progetto.

 Important

Aggiornare il progetto sorgente potrebbe influire sulla precisione dei badge di compilazione del progetto.

5. In Environment (Ambiente):

Per Environment image (Immagine ambiente), procedere in uno dei seguenti modi:

- Per utilizzare un'immagine Docker gestita da AWS CodeBuild, scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Immagine e Versione dell'immagine. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).
 - Per utilizzare un'altra immagine Docker, selezionare Custom image (Immagine personalizzata). Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS
 - Per utilizzare un'immagine Docker privata, scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .
6. In Service role (Ruolo del servizio), procedere in uno dei seguenti modi:
- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.
 - Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

 Note

Quando si utilizza la console per creare o aggiornare un progetto di build, è possibile creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

7. In Buildspec, esegui una delle seguenti operazioni:
- Scegliete Usa un file buildspec per usare il file buildspec.yml nella directory principale del codice sorgente.

- Scegli Inserisci comandi di compilazione per utilizzare la console per inserire i comandi di compilazione.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

8. In Artifacts (Artefatti), per Type (Tipo), procedere in uno dei seguenti modi:

- Se non si desidera creare artefatti di output di compilazione, scegliere No artifacts (Nessun artefatto).
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Per impostazione predefinita, il nome dell'artefatto è quello del progetto. Se si desidera utilizzare un nome diverso, immetterlo nella casella del nome degli artefatti. Se si desidera eseguire l'output di un file .ZIP, includere l'estensione .zip.
 - Per Bucket name (Nome bucket) selezionare il nome del bucket di output.
 - Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, `appspect.yml`, `target/my-app.jar`). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).

9. Espandere Additional configuration (Configurazione aggiuntiva) e scegliere le opzioni come opportuno.

10. Scegliere Create build project (Crea progetto di compilazione). Nella pagina Review (Verifica), selezionare Start build (Avvia compilazione) per eseguire la compilazione.

Accedi ai badge di costruzione AWS CodeBuild

Puoi utilizzare la AWS CodeBuild console o accedere AWS CLI ai badge di costruzione.

- Nella CodeBuild console, nell'elenco dei progetti di compilazione, nella colonna Nome, scegli il link che corrisponde al progetto di build. Nella *project-name* pagina Crea progetto:, in Configurazione, scegli Copia l'URL del badge. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un progetto di compilazione \(console\)](#).

- Nella AWS CLI, esegui il `batch-get-projects` comando. L'URL del badge di compilazione è incluso nella sezione dei dettagli dell'ambiente di progetto dell'output. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un progetto di compilazione \(AWS CLI\)](#).

L'URL di richiesta del badge di compilazione viene generato con un ramo predefinito comune, ma puoi specificare qualsiasi ramo nel tuo repository di origine che hai utilizzato per eseguire una build. Esempio:

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

Puoi anche specificare un tag dal tuo repository di origine sostituendo il branch parametro con il parametro nell'URL del badge tag. Esempio:

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

Pubblica badge di costruzione CodeBuild

Puoi visualizzare lo stato dell'ultima build in un file markdown usando l'URL del tuo build badge in un'immagine markdown. Ciò è utile per visualizzare lo stato della build più recente nel file `readme.md` nel repository dei sorgenti (ad esempio, o). GitHub CodeCommit Esempio:

```

```

CodeBuild stati dei badge

Il badge CodeBuild build può avere uno dei seguenti stati.

- **PASSING (PASSAGGIO)** La più recente compilazione sul ramo determinato è passata.
- **FAILING (IN ERRORE)** La più recente compilazione sul ramo determinato è scaduta, non è riuscita, ha problemi o è stata interrotta.
- **IN_PROGRESS (IN CORSO)** La più recente compilazione nel ramo determinato è in corso.
- **UNKNOWN (SCONOSCIUTO)** Il progetto non ha ancora eseguito una compilazione per il determinato ramo o per niente. Inoltre, la funzionalità dei badge di compilazione potrebbe essere stata disabilitata.

«Rapporto di prova utilizzando il AWS CLI» campione

I test specificati nel file `buildspec` vengono eseguiti durante la compilazione. Questo esempio mostra come utilizzare il per AWS CLI incorporare i test nelle build in CodeBuild. È possibile utilizzare JUnit per creare test unitari oppure utilizzare un altro strumento per creare test di configurazione. La successiva valutazione dei risultati dei test consentirà di risolvere eventuali problemi od ottimizzare l'applicazione.

Puoi utilizzare l' CodeBuild API o la AWS CodeBuild console per accedere ai risultati del test. In questo esempio viene illustrato come configurare il report in modo che i risultati dei test vengano esportati in un bucket S3.

Argomenti

- [Esegui l'esempio del rapporto di prova](#)

Esegui l'esempio del rapporto di prova

Utilizza i seguenti passaggi per eseguire l'esempio del rapporto di prova.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un gruppo di report](#)
- [Fase 2: Configurare un progetto con un gruppo di report](#)
- [Fase 3: Esecuzione e visualizzazione dei risultati di un report](#)

Prerequisiti

- Crea i casi di test. Questo esempio presuppone l'esistenza di casi di test da includere nel report di test esemplificativo. Ricorda di specificare la posizione dei file di test nel file `buildspec`.

Sono supportati i seguenti formati di file di report di test:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNg XML (.xml)

- Visual Studio TRX (.trx)
- Codice XML di Visual Studio TRX (.xml)

Crea i tuoi casi di test con qualsiasi framework di test in grado di creare file di report in uno di questi formati (ad esempio, il JUnit plug-in Surefire, TestNG o Cucumber).

- Crea un bucket S3 e annotane il nome. Per ulteriori informazioni, vedi [Come posso creare un bucket S3?](#) nella Guida per l'utente di Amazon S3.
- Crea un ruolo IAM e prendi nota del relativo ARN. Ti servirà in procinto di creare il progetto di compilazione.
- Se il ruolo non dispone delle autorizzazioni seguenti, aggiungile.

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

Per ulteriori informazioni, consulta [Autorizzazioni per le operazioni correlate ai report di test](#).

Fase 1: Creare un gruppo di report

1. Crea un file denominato `CreateReportGroupInput.json`.
2. Creare una cartella nel bucket S3 in cui vengano esportati i risultati dei test.
3. Copiare quanto segue in `CreateReportGroupInput.json`. Per *<bucket-name>*, utilizzare il nome del bucket S3. Per *<path-to-folder>*, immettere il percorso della cartella nel bucket S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
```



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

4. Eseguire il comando sottostante nella directory che contiene `CreateReportGroupInput.json`.

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

L'output sarà simile al seguente. Annota l'ARN per `reportGroup`, per poterlo utilizzare nel momento in cui verrà creato un progetto a esso correlato.

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

Fase 2: Configurare un progetto con un gruppo di report

Per eseguire un report, devi prima creare un progetto di CodeBuild compilazione configurato con il tuo gruppo di report. I casi di test specificati per il gruppo di report vengono eseguiti in concomitanza di una compilazione.

1. Creare un file buildspec denominato `buildspec.yml`.
2. Utilizzare il seguente YAML come modello per il file `buildspec.yml`. Bisogna includere i comandi che eseguono i test. Nella sezione `reports`, vanno specificati i file contenenti i risultati dei casi di test. Questi file memorizzano i risultati dei test a cui puoi accedere CodeBuild, e scadono 30 giorni dopo la creazione. Inoltre, differiscono dai file con i risultati dei casi di test non elaborati che vengono esportati in un bucket S3.

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: openjdk8
  build:
    commands:
      - echo Running tests
      - <enter commands to run your tests>

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

Note

Invece dell'ARN di un gruppo di report esistente, è possibile specificare un nome per un gruppo di report non ancora creato. Se specifichi un nome anziché un ARN, CodeBuild crea un gruppo di report quando esegue una build. Il nome comprenderà il nome del progetto e quello specificato nel file buildspec in questo formato: `project-name-report-group-name`. Per ulteriori informazioni, consultare [Crea report di test](#) e [Denominazione dei gruppi di report](#).

3. Crea un file denominato `project.json`. Tale file contiene l'input per il comando `create-project`.

4. Copiare il seguente JSON in `project.json`. Per `source`, immettere il tipo e percorso del repository contenente i file di origine. Per `serviceRole`, specificare l'ARN del ruolo in uso.

```
{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|
NO_SOURCE",
    "location": "<your-source-url>"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "small"
  },
  "serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-
name>"
}
```

5. Eseguire il comando sottostante nella directory che contiene `project.json`. Viene così creato un progetto denominato `test-project`.

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

Fase 3: Esecuzione e visualizzazione dei risultati di un report

In questa sezione, viene eseguita una compilazione del progetto creato in precedenza. Durante il processo di compilazione, CodeBuild crea un rapporto con i risultati dei casi di test. Il report è incluso nel gruppo di report specificato.

1. Per avviare una build, esegui il comando seguente. `test-report-project` è il nome del progetto di compilazione creato sopra. prendendo poi nota dell'ID di compilazione visualizzato nell'output.

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

2. Per ottenere informazioni sulla compilazione, incluso l'ARN del report, eseguire il comando sottostante. Per *<build-id>*, specificare l'ID di compilazione. Prendi nota del rapporto ARN nella `reportArns` proprietà dell'output.

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

3. Esegui il comando seguente per ottenere dettagli sul rapporto. Per *<report-arn>*, specificare l'ARN del report.

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

L'output sarà simile al seguente. Questo output di esempio mostra quanti dei test hanno avuto esito positivo, non sono riusciti, sono stati ignorati e hanno generato un errore o restituito uno stato sconosciuto.

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

```

    "durationInNanoSeconds": 6657770,
    "total": 11,
    "statusCounts": {
      "FAILED": 3,
      "SKIPPED": 7,
      "ERROR": 0,
      "SUCCEEDED": 1,
      "UNKNOWN": 0
    }
  }
],
"reportsNotFound": []
}

```

4. Eseguire il comando sottostante per elencare le informazioni sui casi di test del report. Per **<report-arn>**, specificare l'ARN del report. Per il parametro facoltativo **--filter**, è possibile specificare un risultato di stato (SUCCEEDED, FAILED, SKIPPED, ERROR o UNKNOWN).

```

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

```

L'output sarà simile al seguente.

```

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

```

```
"prefix": "Cucumber tests for agent",
"message": "A test message",
"durationInNanoSeconds": 1540540,
"testRawDataPath": "<path-to-output-report-files>"
}
]
}
```

Esempi Docker per CodeBuild

Questa sezione descrive esempi di integrazioni tra Docker e AWS CodeBuild

Project N.E.M.O.	Description
Docker in un esempio di immagine personalizzato per CodeBuild	Questo esempio crea ed esegue un'immagine Docker utilizzando CodeBuild un'immagine di build Docker personalizzata (docker:dind in Docker Hub).
Esempio di server di creazione di immagini Docker per CodeBuild	Questo esempio scarica le build Docker su un server di creazione di immagini gestito.
Windows Docker crea un esempio per CodeBuild	Questo esempio crea ed esegue un'immagine Windows Docker utilizzando CodeBuild
Esempio di «Pubblica immagine Docker in un archivio di immagini Amazon ECR» per CodeBuild	Questo esempio produce come output di build un'immagine Docker e quindi invia l'immagine e Docker a un repository di immagini Amazon Elastic Container Registry (Amazon ECR).
Registro privato con Gestione dei segreti AWS esempio per CodeBuild	Questo esempio mostra come utilizzare un'immagine Docker archiviata in un registro privato come ambiente di runtime. CodeBuild

Docker in un esempio di immagine personalizzato per CodeBuild

L'esempio seguente crea ed esegue un'immagine Docker utilizzando AWS CodeBuild un'immagine di build Docker personalizzata (docker:dind in Docker Hub).

Per scoprire come creare un'immagine Docker utilizzando invece un'immagine di build fornita CodeBuild con il supporto Docker, consulta il nostro. [Esempio «Pubblica immagine Docker su Amazon ECR»](#)

 Important

L'esecuzione di questo esempio potrebbe comportare addebiti sul tuo AWS account. Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon S3 e AWS KMS CloudWatch Logs. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i prezzi](#) di [Amazon S3, i prezzi](#) e [AWS Key Management Service i prezzi](#) di [Amazon CloudWatch](#).

Argomenti

- [Esegui Docker in un esempio di immagine personalizzato](#)

Esegui Docker in un esempio di immagine personalizzato

Usa la seguente procedura per eseguire il Docker in un esempio di immagine personalizzato. Per ulteriori informazioni su questo esempio, vedere [Docker in un esempio di immagine personalizzato per CodeBuild](#).

Per eseguire Docker in un esempio di immagine personalizzato

1. Crea i file come descritto nelle [File](#) sezioni [Struttura delle directory](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository AWS CodeCommit GitHub, o Bitbucket.

 Important

Non caricare *(root directory name)*, ma solo i file all'interno di *(root directory name)*

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere *(root directory name)* al file ZIP, ma solo i file all'interno di *(root directory name)*

2. Crea un progetto di compilazione, esegui la build e visualizza le relative informazioni sulla build.

Se usi il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del `create-project` comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

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

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

3. Per vedere i risultati della compilazione, cercare la stringa `Hello, World!` nel log di compilazione. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli delle compilazioni](#).

Struttura delle directory

Questo esempio assume la seguente struttura delle directory.

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



```
### Dockerfile
```

File

L'immagine di base del sistema operativo utilizzata in questo esempio è di Ubuntu. L'esempio utilizza i seguenti file.

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

```
version: 0.2

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

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

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Esempio di server di creazione di immagini Docker per CodeBuild

L'esempio seguente scarica le build Docker su un server di creazione di immagini gestito. Puoi adattare questo esempio per fornire un server di creazione di immagini Docker dedicato e gestito nella configurazione del tuo progetto. CodeBuild Tieni presente che l'istanza fornita è attiva mentre le build vengono eseguite attivamente per il progetto e viene interrotta quando le build non sono in esecuzione. L'istanza fornita viene archiviata per un massimo di un mese prima di essere riciclata. Per ulteriori informazioni, consulta [CodeBuild Docker](#) Server Capability.

Important

L'esecuzione di questo esempio potrebbe comportare addebiti sul tuo AWS account. Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon S3

e AWS KMS CloudWatch Logs. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i prezzi](#) di [Amazon S3, i prezzi](#) e [AWS Key Management Service i prezzi](#) di [Amazon CloudWatch](#).

Argomenti

- [Configura un server Docker](#)

Configura un server Docker

Utilizza la seguente procedura per fornire un ambiente di calcolo dedicato per un CodeBuild progetto che gestisce i carichi di lavoro Docker e archivia i livelli di immagine Docker.

Per configurare un server Docker

1. Crea i file come descritto nelle [File](#) sezioni [Struttura delle directory](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository AWS CodeCommit GitHub, o Bitbucket.

Important

Non caricare *(root directory name)*, ma solo i file all'interno di *(root directory name)*

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere *(root directory name)* al file ZIP, ma solo i file all'interno di *(root directory name)*

2. Crea un progetto di compilazione, esegui la build e visualizza le relative informazioni sulla build:
 - a. Nella sezione Ambiente della console, scegli Configurazione aggiuntiva, vai alla configurazione del server Docker, quindi seleziona Abilita il server docker per questo progetto. Puoi quindi scegliere il tipo di elaborazione del server Docker e fornire una credenziale di registro.
 - b. Se si utilizza il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del comando potrebbe essere simile a questo. `create-project` (Sostituire i segnaposto con i propri valori).

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

```

    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerServerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:5.0",
    "computeType": "BUILD_GENERAL1_LARGE",
    "dockerServer": [
      {
        "computeType": "BUILD_GENERAL1_LARGE",
        "securityGroupIds": [ "security-groups-ID" ]
      }
    ]
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}

```

Note

I gruppi di sicurezza configurati per i server Docker dovrebbero consentire il traffico di rete in ingresso dal VPC configurato nel progetto. Dovrebbero consentire l'ingresso sulla porta 9876.

3. Per vedere i risultati della compilazione, cercare la stringa `Hello, World!` nel log di compilazione. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli delle compilazioni](#).

Struttura delle directory

Questo esempio assume la seguente struttura delle directory.

```

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

```

File

L'immagine di base del sistema operativo utilizzata in questo esempio è di Ubuntu. L'esempio utilizza i seguenti file.

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

```
version: 0.2

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

Dockerfile (in *(root directory name)*)

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

RUN echo "Hello World"
```

Windows Docker crea un esempio per CodeBuild

L'esempio seguente crea ed esegue un'immagine di Windows Docker utilizzando CodeBuild

Argomenti

- [Esempio di build di Esegui Windows Docker](#)

Esempio di build di Esegui Windows Docker

Utilizzare la procedura seguente per eseguire le build di Windows Docker.

Per eseguire le build di Windows Docker, esempio

1. Crea i file come descritto nelle [File](#) sezioni [Struttura delle directory](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository AWS CodeCommit GitHub, o Bitbucket.

Important

Non caricare *(root directory name)*, ma solo i file all'interno di *(root directory name)*

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere (*root directory name*) al file ZIP, ma solo i file all'interno di (*root directory name*)

2. Crea una flotta. WINDOWS_EC2

Se usi il AWS CLI per creare la flotta, l'input in formato JSON del `create-fleet` comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

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

3. Crea un progetto di compilazione, esegui la build e visualizza le informazioni relative alla build.

Se usi il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del `create-project` comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

```
{
  "name": "project-name",
  "source": {
    "type": "S3",
    "location": "bucket-name/DockerImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "WINDOWS_EC2",
    "image": "Windows",
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "fleet": {
      "fleetArn": "fleet-arn"
    }
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}
```

4. Per vedere i risultati della compilazione, cercare la stringa Hello, World! nel log di compilazione. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli delle compilazioni](#).

Struttura delle directory

Questo esempio assume la seguente struttura delle directory.

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

File

L'immagine di base del sistema operativo utilizzato in questo esempio è `mcr.microsoft.com/windows/servercore:ltsc2022`. L'esempio utilizza i seguenti file.

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

```
version: 0.2

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

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

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

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

Esempio di «Pubblica immagine Docker in un archivio di immagini Amazon ECR» per CodeBuild

Questo esempio produce come output di build un'immagine Docker e quindi invia l'immagine Docker a un repository di immagini Amazon Elastic Container Registry (Amazon ECR). Questo esempio può

essere modificato per inviare l'immagine Docker a Docker Hub. Per ulteriori informazioni, consulta [Adatta l'esempio «Pubblica immagine Docker su Amazon ECR» per inviarlo a Docker Hub](#).

Per imparare a compilare un'immagine Docker utilizzando un'immagine Docker di compilazione personalizzata, (`docker:dind` in Docker Hub), consulta [Docker in un esempio di immagine personalizzata](#).

Questo esempio è stato testato sulla base di `go1.12`.

Questo esempio utilizza la nuova funzione di compilazioni Docker a più fasi, che produce un'immagine Docker come output della compilazione. Quindi invia l'immagine Docker a un repository di immagini Amazon ECR. Le compilazioni di immagini Docker a più fasi aiutano a ridurre la dimensione dell'immagine Docker finale. Per ulteriori informazioni, consultare la pagina relativa all'[utilizzo di compilazioni a più fasi con Docker](#).

Important

L'esecuzione di questo esempio potrebbe comportare addebiti sul tuo account. AWS Questi includono possibili addebiti per AWS CodeBuild e per AWS risorse e azioni relative ad Amazon S3 AWS KMS, CloudWatch Logs e Amazon ECR. Per ulteriori informazioni, consulta [CodeBuild i prezzi, i prezzi di Amazon S3, i prezzi, i AWS Key Management Service prezzi di Amazon e CloudWatch i prezzi di Amazon Elastic Container Registry](#).

Argomenti

- [Esegui l'esempio «Pubblica immagine Docker su Amazon ECR»](#)
- [Adatta l'esempio «Pubblica immagine Docker su Amazon ECR» per inviarlo a Docker Hub](#)

Esegui l'esempio «Pubblica immagine Docker su Amazon ECR»

Utilizza la procedura seguente per eseguire l'esempio che pubblica un'immagine Docker su Amazon ECR. Per ulteriori informazioni su questo esempio, consulta [Esempio di «Pubblica immagine Docker in un archivio di immagini Amazon ECR» per CodeBuild](#)

Per eseguire questo esempio

1. Se disponi già di un repository di immagini in Amazon ECR che desideri utilizzare, vai al passaggio 3. Altrimenti, se utilizzi un utente anziché un account AWS root o un utente

amministratore per lavorare con Amazon ECR, aggiungi questa istruzione (tra **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) all'utente (o al gruppo IAM a cui è associato l'utente). L'utilizzo di un account AWS root non è consigliato. Questa istruzione consente la creazione di repository Amazon ECR per l'archiviazione di immagini Docker. Le ellissi (. . .) vengono utilizzate per brevità e per aiutare i clienti a individuare il punto in cui aggiungere l'istruzione. Non rimuovere nessuna istruzione e non digitare queste ellissi nella policy. Per ulteriori informazioni, consulta [Working with inline policies](#) using the in the user Guide. Console di gestione AWS

JSON

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

Note

L'entità IAM che modifica questa policy deve avere l'autorizzazione in IAM a modificare le policy.

2. Crea un archivio di immagini in Amazon ECR. Assicurati di creare il repository nella stessa AWS regione in cui crei l'ambiente di compilazione ed esegui la build. Per ulteriori informazioni, consulta [Creazione di un repository](#) nella Amazon ECR User Guide. Il nome di questo repository deve essere uguale a quello che viene specificato più avanti in questa procedura, rappresentato dalla variabile di ambiente IMAGE_REPO_NAME. Assicurati che la policy del repository Amazon ECR conceda l'accesso tramite push alle immagini per il tuo ruolo IAM di CodeBuild servizio.
3. Aggiungi questa dichiarazione (tra **### BEGIN ADDING STATEMENT HERE ###** e **### END ADDING STATEMENT HERE ###**) alla policy che hai allegato al tuo AWS CodeBuild ruolo di

servizio. Questa dichiarazione consente di CodeBuild caricare immagini Docker negli archivi Amazon ECR. Le ellissi (. . .) vengono utilizzate per brevità e per aiutare i clienti a individuare il punto in cui aggiungere l'istruzione. Non rimuovere nessuna istruzione e non digitare queste ellissi nella policy.

JSON

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

Note

L'entità IAM che modifica questa policy deve avere l'autorizzazione in IAM per modificare le policy.

4. Crea i file come descritto nelle [File](#) sezioni [Struttura delle directory](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository AWS CodeCommit GitHub, o Bitbucket. Per ulteriori informazioni, consulta il [riferimento al file delle definizioni delle immagini](#) nella Guida per l'utente.AWS CodePipeline

Important

Non caricare (*root directory name*), ma solo i file all'interno di (*root directory name*)

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere (*root directory name*) al file ZIP, ma solo i file all'interno di (*root directory name*)

5. Crea un progetto di compilazione, esegui la build e visualizza le informazioni sulla build.

Se utilizzi la console per creare il progetto:

- a. In Operating system (Sistema operativo), seleziona Ubuntu.
- b. In Runtime, seleziona Standard.
- c. Per Image, scegliere: `aws/codebuild/standard:5.0`.
- d. Aggiungere le seguenti variabili di ambiente:
 - `AWS_DEFAULT_REGION` con un valore di *region-ID*
 - `AWS_ACCOUNT_ID` con un valore di *account-ID*
 - `IMAGE_TAG` con un valore di Latest
 - `IMAGE_REPO_NAME` con un valore di *Amazon-ECR-repo-name*

Se si utilizza il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del comando potrebbe avere un aspetto simile a questo. `create-project` (Sostituire i segnaposto con i propri valori).

```
{
  "name": "sample-docker-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      }
    ]
  }
}
```

```

    },
    {
      "name": "AWS_ACCOUNT_ID",
      "value": "account-ID"
    },
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ],
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

6. Conferma di aver CodeBuild inviato correttamente l'immagine Docker al repository:

1. Apri la console Amazon ECR all'indirizzo <https://console.aws.amazon.com/ecr/>.
2. Selezionare il nome del repository L'immagine deve essere elencata nella colonna Tag immagine.

Struttura delle directory

Questo esempio assume la seguente struttura delle directory.

```

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

```

File

Questo esempio utilizza i seguenti file.

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

```
version: 0.2
```

```

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

```

Dockerfile (in *(root directory name)*)

```

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

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

```

Note

CodeBuild sostituisce le immagini Docker ENTRYPOINT personalizzate.

Adatta l'esempio «Pubblica immagine Docker su Amazon ECR» per inviarlo a Docker Hub

Per adattare l'esempio 'Publish Docker image to Amazon ECR' in modo che l'immagine Docker venga inviata a Docker Hub anziché ad Amazon ECR, modifica il codice dell'esempio. Per ulteriori informazioni sull'esempio, consulta e. [Esempio di «Pubblica immagine Docker in un archivio di immagini Amazon ECR» per CodeBuild](#) [Esegui l'esempio «Pubblica immagine Docker su Amazon ECR»](#)

Note

Se si utilizza una versione di Docker precedente alla 17.06, rimuovere l'opzione `--no-include-email`.

1. Sostituisci queste righe di codice specifiche di Amazon ECRS nel `buildspec.yml` file:

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

con queste righe di codice specifiche di Docker Hub:

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

2. Carica il codice modificato in un bucket di input S3 o in un repository AWS CodeCommit GitHub, o Bitbucket.

Important

Non caricare (*root directory name*), ma solo i file all'interno di (*root directory name*)

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere (*root directory name*) al file ZIP, ma solo i file all'interno di (*root directory name*)

3. Sostituire queste righe di codice dall'input in formato JSON al comando create-project:

```
...
"environmentVariables": [
  {
    "name": "AWS_DEFAULT_REGION",
    "value": "region-ID"
  },
  {
    "name": "AWS_ACCOUNT_ID",
    "value": "account-ID"
  },
]
```

```
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ]
  ...
```

con queste righe di codice:

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

4. Crea un ambiente di compilazione, esegui la build e visualizza le informazioni relative alla build.
5. Conferma di AWS CodeBuild aver inviato correttamente l'immagine Docker al repository. Accedere a Docker Hub, andare nel repository e selezionare la scheda Tags (Tag). Il tag `latest` dovrebbe contenere un valore di Last Updated (Ultimo aggiornamento) molto recente.

Registro privato con Gestione dei segreti AWS esempio per CodeBuild

Questo esempio mostra come utilizzare un'immagine Docker archiviata in un registro privato come ambiente AWS CodeBuild di runtime. Le credenziali del registro privato sono archiviate in Gestione dei segreti AWS. Qualsiasi registro privato funziona con CodeBuild. Questo esempio si basa su Docker Hub.

 Note

I segreti sono visibili alle azioni e non vengono mascherati quando vengono scritti su un file.

Argomenti

- [Requisiti per un esempio di registro privato](#)
- [Crea un CodeBuild progetto con un registro privato](#)
- [Configura una credenziale di registro privata per i corridori ospitati autonomamente](#)

Requisiti per un esempio di registro privato

Per utilizzare un registro privato con AWS CodeBuild, è necessario disporre di quanto segue:

- Un segreto di Secrets Manager che memorizza le credenziali del Docker Hub. Le credenziali vengono utilizzate per accedere al repository privato.

 Note

Ti verranno addebitati i segreti che crei.

- Un repository o un account privato.
- Una policy IAM relativa ai ruoli di CodeBuild servizio che concede l'accesso al segreto di Secrets Manager.

Segui questi passaggi per creare queste risorse e quindi creare un progetto di CodeBuild compilazione utilizzando le immagini Docker archiviate nel tuo registro privato.

Crea un CodeBuild progetto con un registro privato

1. Per informazioni su come creare un repository privato gratuito, consulta la pagina relativa ai [repository su Docker Hub](#). Puoi anche eseguire i seguenti comandi in un terminale per estrarre un'immagine, ottenerne l'ID e inserirla in un nuovo repository.

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



```
docker login
docker push your-username/repository-name
```

2. Segui la procedura descritta in [Creare un Gestione dei segreti AWS segreto](#) nella Guida Gestione dei segreti AWS per l'utente.
 - a. Nel passaggio 3, in Scegli il tipo di segreto, scegli Altro tipo di segreto.
 - b. Nelle coppie chiave/valore, crea una coppia chiave-valore per il nome utente di Docker Hub e una coppia chiave-valore per la password di Docker Hub.
 - c. [Continua a seguire i passaggi descritti in Creare un segreto. Gestione dei segreti AWS](#)
 - d. Nel passaggio 5, nella pagina Configura rotazione automatica, disattivala perché le chiavi corrispondono alle credenziali del tuo Docker Hub.
 - e. Completa la procedura descritta in [Creare un Gestione dei segreti AWS segreto](#).

Per maggiori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#)

3. Quando crei un AWS CodeBuild progetto nella console, CodeBuild assegna l'autorizzazione richiesta. Se si utilizza una AWS KMS chiave diversa da `DefaultEncryptionKey`, è necessario aggiungerla al ruolo di servizio. Per ulteriori informazioni, consulta [Modificare un ruolo \(console\)](#) nella Guida per l'utente IAM.

Affinché il ruolo di servizio funzioni con Secrets Manager, deve disporre almeno dell'`secretsmanager:GetSecretValue` autorizzazione.

▼ KMS (1 action) Clone Remove

Service KMS

Actions Write
Decrypt

Resources ☒ Specific ☐ All resources close

key ? EDIT Any

[Add ARN to restrict access](#)

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

4. Per utilizzare la console per la creazione di un progetto in un ambiente archiviato in un registro privato, esegui la seguente procedura durante l'operazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

Note

Se il registro privato si trova nel tuo VPC, deve disporre di un accesso pubblico a Internet. CodeBuild non è possibile estrarre un'immagine da un indirizzo IP privato in un VPC.

- a. In Immagine dell'ambiente, scegli Immagine personalizzata.
- b. In Environment type (Tipo ambiente), seleziona Linux o Windows.
- c. Per Registro delle immagini, scegli Altro registro.
- d. In URL di registro esterno, inserisci la posizione dell'immagine e in Credenziali di registro (facoltativo) inserisci l'ARN o il nome delle tue credenziali di Secrets Manager.

Note

Se le credenziali non esistono nella regione corrente, devi utilizzare l'ARN. Non puoi utilizzare il nome delle credenziali se esistono in un'altra regione.

Configura una credenziale di registro privata per i corridori ospitati autonomamente

Usa le seguenti istruzioni per configurare una credenziale di registro per un runner ospitato autonomamente.

Note

Tieni presente che queste credenziali verranno utilizzate solo se le immagini vengono sostituite con quelle dei registri privati.

AWS Management Console

1. [Apri la console su codebuild/home. AWS CodeBuild https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Crea un progetto di compilazione o seleziona un progetto esistente. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).

3. In Ambiente, scegli Configurazione aggiuntiva.
4. In Configurazione aggiuntiva, immettere il nome o l'ARN del codice segreto Gestione dei segreti AWS per la credenziale del Registro di sistema (facoltativo).

Registry credential - *optional*

AWS CLI

1. Se desideri creare un nuovo progetto, esegui il comando create-project.

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

2. Se desideri aggiornare un progetto esistente, esegui il comando update-project.

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

Creazione di un sito Web statico con output di compilazione ospitato in un bucket S3

È possibile disabilitare la crittografia degli artefatti in una compilazione. Potresti volere eseguire questa operazione per pubblicare gli artefatti in una posizione configurata per l'hosting di un sito Web (non è possibile pubblicare artefatti crittografati). Questo esempio illustra come utilizzare webhook per attivare una compilazione e pubblicare i suoi artefatti in un bucket S3 configurato come sito Web.

1. Seguire le istruzioni nell'argomento relativo alla [configurazione di un sito Web statico](#) per configurare un bucket S3 con funzione di sito Web.

2. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
3. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
4. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
5. In Source (Origine), per Source provider (Fornitore origine), scegliere GitHub. Segui le istruzioni per connetterti (o riconnetterti) con GitHub, quindi scegli Autorizza.

Per Webhook, selezionare Rebuild every time a code change is pushed to this repository (Ricompile ogni volta che viene inviata una modifica del codice a questo repository). È possibile selezionare questa casella solo se si sceglie Use a repository in my account (Usa un repository nel mio account).

SourceAdd source

Source 1 - Primary

Source provider

GitHub

Repository

☐ Public repository

☒ Repository in my GitHub account

GitHub repository

Disconnect GitHub account

▼ Additional configuration

Git clone depth

Git clone depth - optional

1

Build Status - optional

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

Webhook - optional

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

Branch filter - optional

Enter a regular expression

6. In Environment (Ambiente):

Per Environment image (Immagine ambiente), procedere in uno dei seguenti modi:

- Per utilizzare un'immagine Docker gestita da AWS CodeBuild, scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Immagine e Versione dell'immagine. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).

- Per utilizzare un'altra immagine Docker, selezionare Custom image (Immagine personalizzata). Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS
- Per utilizzare un'immagine Docker privata, scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

7. In Service role (Ruolo del servizio), procedere in uno dei seguenti modi:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.
- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

Note

Quando utilizzi la console per creare o aggiornare un progetto di build, puoi creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

8. In Buildspec, esegui una delle seguenti operazioni:

- Scegliete Usa un file buildspec per usare il file buildspec.yml nella directory principale del codice sorgente.
- Scegli Inserisci comandi di compilazione per utilizzare la console per inserire i comandi di compilazione.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

9. In Artifacts, per Type, scegli Amazon S3 per archiviare l'output della build in un bucket S3.
10. Per Bucket name (Nome del bucket), selezionare il nome del bucket S3 precedentemente configurato come sito Web nella fase 1.
11. Se si sceglie Insert build commands (Inserisci comandi di compilazione) in Environment (Ambiente), allora per Output files (File di output) immettere i percorsi dei file della compilazione che si desidera inserire nel bucket di output. Se si dispone di più di un percorso, utilizzare una virgola per separare ogni percorso (ad esempio **appspec.yml, target/my-app.jar**). Per ulteriori informazioni, consulta [Artifacts reference-key in the buildspec file](#).
12. Selezionare Disable artifacts encryption (Disabilita crittografia artefatti).
13. Espandere Additional configuration (Configurazione aggiuntiva) e scegliere le opzioni come opportuno.
14. Scegliere Create build project (Crea progetto di compilazione). Nella pagina del progetto di compilazione, nella sezione Build history (Cronologia compilazione), selezionare Start build (Avvia la compilazione) per eseguire la compilazione.
15. (Facoltativo) Segui le istruzioni riportate in [Esempio: velocizza il tuo sito Web con Amazon S3 CloudFront](#) nella Amazon S3 Developer Guide.

Diverse origini di input ed esempi di artefatti di output

È possibile creare un progetto di AWS CodeBuild compilazione con più di una sorgente di input e più di un set di artefatti di output. Questo esempio illustra come configurare un progetto di compilazione che:

- Utilizza più origini e repository di tipi diversi.
- Pubblica artefatti di compilazione in più bucket S3 in un'unica compilazione.

Nell'esempio seguente, create un progetto di compilazione e lo utilizzate per eseguire una build. L'esempio utilizza il file buildspec del progetto di compilazione per dimostrare come integrare diverse origini e creare più di un set di artefatti.

Per informazioni su come creare una pipeline che utilizza più input di origine per creare più CodeBuild artefatti di output, consulta. [Esempio di CodePipeline/CodeBuild integrazione con più sorgenti di input e artefatti di output](#)

Argomenti

- [Crea un progetto di compilazione con più input e output](#)
- [Crea un progetto di compilazione senza una fonte](#)

Crea un progetto di compilazione con più input e output

Utilizzate la procedura seguente per creare un progetto di compilazione con più input e output.

Per creare un progetto di compilazione con più input e output

1. Carica le tue fonti in uno o più bucket S3, CodeCommit GitHub, GitHub Enterprise Server o Bitbucket.
2. Scegli quale origine è quella principale. Questa è la fonte in cui CodeBuild cerca ed esegue il file buildspec.
3. Creare un progetto di compilazione. Per ulteriori informazioni, consulta [Crea un progetto di compilazione in AWS CodeBuild](#).
4. Crea il tuo progetto di build, esegui la build e ottieni informazioni sulla build.
5. Se usi il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del create-project comando potrebbe essere simile al seguente:

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [ss
    {
```



```
    "type": "S3",
    "location": "<output-bucket>",
    "artifactIdentifier": "artifact1"
  },
  {
    "type": "S3",
    "location": "<other-output-bucket>",
    "artifactIdentifier": "artifact2"
  }
],
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

L'origine primaria viene definita dall'attributo `source`. Tutte le altre origini sono chiamate origini secondarie e appaiono sotto `secondarySources`. Tutte le origini secondarie sono installate nella propria directory. Tale directory è archiviata nella variabile di ambiente integrata `CODEBUILD_SRC_DIR_`*sourceIdentifier*. Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).

L'attributo `secondaryArtifacts` contiene un elenco di definizioni di artefatti. Tali artefatti utilizzano il blocco `secondary-artifacts` del file `buildspec` presente all'interno del blocco `artifacts`.

Gli artefatti secondari nel file `buildspec` hanno la stessa struttura degli artefatti e sono separati dall'identificatore degli artefatti.

Note

Nell'[API CodeBuild](#), `artifactIdentifier` nell'artefatto secondario è un attributo richiesto in `CreateProject` e `UpdateProject`. Deve essere utilizzato per fare riferimento a un secondo artefatto.

Se si utilizza l'input precedente in formato JSON, il file `buildspec` del progetto potrebbe risultare simile a questo:

```
version: 0.2

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

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

Puoi sovrascrivere la versione dell'origine primaria utilizzando l'API con l'attributo `sourceVersion` in `StartBuild`. Per sovrascrivere una o più versioni sorgente secondarie, utilizzare l'attributo `secondarySourceVersionOverride`.

L'input in formato JSON per il comando in potrebbe essere simile a: `start-build` AWS CLI

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

```
]
}
```

Crea un progetto di compilazione senza una fonte

Puoi configurare un CodeBuild progetto scegliendo il tipo di **NO_SOURCE** sorgente quando configuri la tua fonte. Quando il tipo di origine è **NO_SOURCE**, non puoi specificare un file buildspec perché il progetto non possiede un'origine. Al contrario, devi specificare una stringa buildspec in formato YAML nell'attributo buildspec dell'input in formato JSON nel comando dell'interfaccia a riga di comando `create-project`. Potrebbe essere simile a quanto segue:

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

Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).

Versioni di runtime nell'esempio di file buildspec per CodeBuild

Se usi l'immagine standard di Amazon Linux 2 (AL2) versione 1.0 o successiva o l'immagine standard di Ubuntu versione 2.0 o successiva, puoi specificare uno o più runtime nella `runtime-versions` sezione del tuo file buildspec. Gli esempi seguenti mostrano come modificare il runtime del progetto, specificare più di un runtime e specificare un runtime che dipende da un altro runtime. Per informazioni sui runtime supportati, vedi [Immagini Docker fornite da CodeBuild](#).

Note

Se utilizzi Docker nel container di compilazione, la compilazione deve essere eseguita in modalità con privilegi. Per ulteriori informazioni, consultare [Esegui AWS CodeBuild le build manualmente](#) e [Crea un progetto di compilazione in AWS CodeBuild](#).

Argomenti

- [Aggiorna la versione di runtime nel file buildspec](#)
- [Specifica di due runtime](#)

Aggiorna la versione di runtime nel file buildspec

Puoi modificare il runtime utilizzato dal tuo progetto con una nuova versione aggiornando la `runtime-versions` sezione del tuo file buildspec. I seguenti esempi mostrano come specificare le versioni 8 e 11 di Java:

- Una sezione `runtime-versions` che specifica la versione 8 di Java:

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

- Una sezione `runtime-versions` che specifica la versione 11 di Java:

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

I seguenti esempi mostrano come specificare diverse versioni di Python usando l'immagine standard di Ubuntu 5.0 o l'immagine standard di Amazon Linux 2 3.0:

- Una `runtime-versions` sezione che specifica la versione 3.7 di Python:

```
phases:
  install:
```

```
runtime-versions:
  python: 3.7
```

- Una `runtime-versions` sezione che specifica la versione 3.8 di Python:

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

Questo esempio illustra un progetto che inizia con il runtime Java versione 8 e viene aggiornato al runtime Java versione 10.

1. Scaricare e installare Maven. Per informazioni, consultare le pagine del sito Web di Apache Maven relative al [download di Apache Maven](#) e all'[installazione di Apache Maven](#).
2. Passare a una directory vuota sull'istanza o sul computer locale, quindi eseguire il comando Maven.

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

Se il comando è eseguito correttamente, vengono creati la struttura e i file della directory.

```
.
### R00T
  ### pom.xml
  ### src
    ### main
      ### resources
      ### webapp
        ### WEB-INF
        #   ### web.xml
        ### index.jsp
```

3. Crea un file denominato `buildspec.yml` con i seguenti contenuti. Archivia il file nella directory *(root directory name)/my-web-app*.

```
version: 0.2

phases:
```

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

Nel file buildspec:

- La sezione `runtime-versions` specifica che il progetto utilizza la versione 8 del runtime Java .
- Il comando `- java -version` visualizza la versione di Java utilizzata dal progetto al momento della compilazione.

La struttura del file dovrebbe avere il seguente aspetto.

```
(root directory name)
### my-web-app
  ### src
  #   ### main
  #   ### resources
  #   ### webapp
  #       ### WEB-INF
  #           ### web.xml
  #               ### index.jsp
  ### buildspec.yml
  ### pom.xml
```

4. Carica il contenuto della `my-web-app` directory in un bucket di input S3 o in un repository CodeCommit, GitHub o Bitbucket.

Important

Non caricare *(root directory name)* o *(root directory name)/my-web-app*, ma solo le directory e i file all'interno di *(root directory name)/my-web-app*.

Se si sta usando un bucket di input S3, assicurarsi di creare un file ZIP che contenga la struttura e i file della directory, quindi caricarlo nel bucket di input. Non aggiungere *(root directory name)* o *(root directory name)/my-web-app* al file ZIP, ma solo le directory e i file all'interno di *(root directory name)/my-web-app*.

5. [Apri la console su codebuild/home. AWS CodeBuild https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
6. Creare un progetto di compilazione. Per ulteriori informazioni, consultare [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#). Lasciare tutte le impostazioni sui valori predefiniti, ad eccezione delle seguenti.
 - In Environment (Ambiente):
 - Per Environment image (Immagine ambiente), scegliere Managed image (Immagine gestita).
 - Per Operating system (Sistema operativo), scegliere Amazon Linux 2.
 - In Runtime(s) (Runtime), seleziona Standard.
 - Per Image, scegli -x86_64-standard:4.0. aws/codebuild/amazonlinux
7. Selezionare Start build (Avvia compilazione).
8. In Build configuration (Configurazione della compilazione) accettare le impostazioni predefinite, quindi scegliere Build configuration (Avvia compilazione).
9. Al termine della compilazione, visualizzare l'output della compilazione nella scheda Build logs (Log di compilazione). Verrà visualizzato un output simile al seguente:

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

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

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

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

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

10. Aggiorna la sezione `runtime-versions` con Java versione 11:

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

11. Dopo aver salvato la modifica, esegui nuovamente la compilazione e visualizza l'output di compilazione. La versione di Java installata visualizzata è 11. Verrà visualizzato un output simile al seguente:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE  
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src  
[Container] Date Time YAML location is /codebuild/output/src460614277/src/  
buildspec.yml  
[Container] Date Time Processing environment variables  
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual  
selections...  
Installing Java version 11 ...  
  
[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"  
  
[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"  
  
[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"  
  
[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*  
"$JRE_11_HOME"/bin/*;
```

Specifica di due runtime

È possibile specificare più di un runtime nello stesso progetto di build. CodeBuild Questo progetto di esempio utilizza due file di origine: uno che utilizza il comando `Vai a runtime` e uno che utilizza il runtime `Node.js`.

1. Crea una directory denominata `my-source`.
2. Nella directory `my-source`, crea una directory denominata `golang-app`.

3. Crea un file denominato `hello.go` con i seguenti contenuti. Archivia il file nella directory `golang-app`.

```
package main
import "fmt"

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

4. Nella directory `my-source`, crea una directory denominata `nodejs-app`. che deve trovarsi allo stesso livello della directory `golang-app`.
5. Crea un file denominato `index.js` con i seguenti contenuti. Archivia il file nella directory `nodejs-app`.

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

6. Crea un file denominato `package.json` con i seguenti contenuti. Archivia il file nella directory `nodejs-app`.

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

```
}
```

7. Crea un file denominato `buildspec.yml` con i seguenti contenuti. Archivia il file nella `my-source` directory, allo stesso livello delle directory `nodejs-app` e `golang-app`. La `runtime-versions` sezione specifica i runtime di Node.js versione 12 e Go versione 1.13.

```
version: 0.2

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

8. La struttura del file dovrebbe avere il seguente aspetto.

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

9. Carica il contenuto della `my-source` directory in un bucket di input S3 o in un repository CodeCommit, GitHub o Bitbucket.

 Important

Se si sta usando un bucket di input S3, assicurarsi di creare un file ZIP che contenga la struttura e i file della directory, quindi caricarlo nel bucket di input. Non aggiungere `my-source` al file ZIP, ma solo le directory e i file all'interno di `my-source`.

10. [Apri la console su codebuild/home. AWS CodeBuild https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
11. Creare un progetto di compilazione. Per ulteriori informazioni, consultare [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#). Lasciare tutte le impostazioni sui valori predefiniti, ad eccezione delle seguenti.
 - In Environment (Ambiente):
 - Per Environment image (Immagine ambiente), scegliere Managed image (Immagine gestita).
 - Per Operating system (Sistema operativo), scegliere Amazon Linux 2.
 - In Runtime(s) (Runtime), seleziona Standard.
 - Per Image, scegli `-x86_64-standard:4.0. aws/codebuild/amazonlinux`
12. Scegliere Create build project (Crea progetto di compilazione).
13. Selezionare Start build (Avvia compilazione).
14. In Build configuration (Configurazione della compilazione) accettare le impostazioni predefinite, quindi scegliere Build configuration (Avvia compilazione).
15. Al termine della compilazione, visualizzare l'output della compilazione nella scheda Build logs (Log di compilazione). Verrà visualizzato un output simile al seguente, che mostra l'output dei runtime Go e Node.js. Inoltre, mostra l'output alle applicazioni Go e Node.js.

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

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

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

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

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

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

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

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

[Container] Date Time Running command npm run test

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

run some tests here
```

Esempio di versione sorgente con AWS CodeBuild

In questo esempio viene illustrato come specificare una versione dell'origine utilizzando un formato diverso da un ID commit (detto anche SHA commit). Puoi specificare la versione dell'origine nei seguenti modi:

- Per un provider di sorgenti Amazon S3, utilizza l'ID di versione dell'oggetto che rappresenta il file ZIP di input della build.
- Per CodeCommit Bitbucket ed GitHub Enterprise Server, utilizza uno dei seguenti: GitHub
 - Richiesta pull come riferimento alla richiesta pull (ad esempio `refs/pull/1/head`).
 - Branch come nome di filiale.
 - ID del commit.
 - Etichetta.
- Riferimento e ID di commit. Il riferimento può essere uno dei seguenti:
 - Un tag (ad esempio `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Un ramo (ad esempio `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Una richiesta pull (ad esempio `refs/pull/1/head^{full-commit-SHA}`).
- Per GitLab e GitLab Self Managed, utilizza uno dei seguenti:
 - Branch come nome del ramo.
 - ID del commit.
 - Etichetta.

Note

È possibile specificare la versione di una sorgente di pull request solo se il repository è GitHub o GitHub Enterprise Server.

Se utilizzi un riferimento e un ID commit per specificare una versione, la fase `DOWNLOAD_SOURCE` della compilazione dura meno e più rapida di quando fornisci solo la versione. Questo perché quando si aggiunge un riferimento, CodeBuild non è necessario scaricare l'intero repository per trovare il commit.

- È possibile specificare una versione di origine con solo un ID commit, ad esempio `12345678901234567890123467890123456789`. Se lo fai, CodeBuild devi scaricare l'intero repository per trovare la versione.
- È possibile specificare una versione di origine con un riferimento e un ID commit in questo formato: `refs/heads/branchname^{full-commit-SHA}` (ad esempio `refs/heads/main^{12345678901234567890123467890123456789}`). Se lo fai, CodeBuild scarica solo il ramo specificato per trovare la versione.

 Note

Per velocizzare la `DOWNLOAD_SOURCE` fase di compilazione, puoi anche impostare la profondità del clone di Git su un numero basso. CodeBuild scarica un numero inferiore di versioni del tuo repository.

Argomenti

- [Specificate una versione GitHub del repository con un ID di commit](#)
- [Specificate una versione del repository con un riferimento e un ID di commit GitHub](#)

Specificate una versione GitHub del repository con un ID di commit

È possibile specificare una versione di origine con solo un ID commit, ad esempio `12345678901234567890123467890123456789`. Se si esegue questa operazione, è CodeBuild necessario scaricare l'intero repository per trovare la versione.

Per specificare una versione del GitHub repository con un ID di commit

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#). Lasciare tutte le impostazioni sui valori predefiniti, ad eccezione delle seguenti:
 - In Source (Origine):
 - Per Source provider, scegli. GitHub Se non sei connesso a GitHub, segui le istruzioni per connetterti.
 - In Repository, scegliere Public repository (Repository pubblico).
 - Nel campo Repository URL (URL repository), inserire **`https://github.com/aws/aws-sdk-ruby.git`**
 - In Environment (Ambiente):
 - Per Environment image (Immagine ambiente), scegliere Managed image (Immagine gestita).
 - Per Operating system (Sistema operativo), scegliere Amazon Linux 2.
 - In Runtime(s) (Runtime), seleziona Standard.
 - Per Image, scegliete `aws/codebuild/amazonlinux-x86_64-standard:4.0`.

3. Per Build specifications (Specifiche di compilazione), scegliere Insert build commands (Inserisci comandi di compilazione) e selezionare Switch to editor (Passa a editor).
4. Per Build commands (Comandi di compilazione), sostituire il testo segnaposto con il seguente:

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  build:
    commands:
      - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

La sezione `runtime-versions` è obbligatoria quando si utilizza l'immagine standard di Ubuntu 2.0. Qui viene specificata la versione 2.6 del runtime di Ruby, ma si può utilizzare qualunque runtime. Il comando `echo` visualizza la versione del codice sorgente archiviato nella variabile di ambiente `CODEBUILD_RESOLVED_SOURCE_VERSION`.

5. In Build configuration (Configurazione della compilazione) accettare le impostazioni predefinite, quindi scegliere Build configuration (Avvia compilazione).
6. Per Source version (Versione origine), inserire **046e8b67481d53bdc86c3f6affdd5d1afae6d369**. Questo è il SHA di un commit nel repository <https://github.com/aws/aws-sdk-ruby.git>.
7. Selezionare Start build (Avvia compilazione).
8. Una volta completata la compilazione, si dovrebbe visualizzare quanto segue:
 - Nella scheda Build logs (Log di compilazione), la versione dell'origine del progetto che è stata utilizzata. Ecco un esempio.

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

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

- Nella scheda Environment variables (Variabili di ambiente), Resolved source version (Versione di origine risolta) corrisponde all'ID commit utilizzato per creare la compilazione.
- Nella scheda Phase details (Dettagli di fase), la durata della fase `DOWNLOAD_SOURCE`.

Specificare una versione del repository con un riferimento e un ID di commit GitHub

È possibile specificare una versione di origine con un riferimento e un ID commit in questo formato: `refs/heads/branchname^{full-commit-SHA}` (ad esempio `refs/heads/main^{12345678901234567890123467890123456789}`). In tal caso, CodeBuild scarica solo il ramo specificato per trovare la versione.

Per specificare una versione del GitHub repository con un riferimento e un ID di commit.

1. Completa le fasi descritte in [Specificare una versione GitHub del repository con un ID di commit](#).
2. Dal riquadro di navigazione a sinistra, scegliere Build projects (Progetti di compilazione) e selezionare il progetto creato in precedenza.
3. Selezionare Start build (Avvia compilazione).
4. In Source version (Versione di origine), inserire `refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}`. Questo è lo stesso ID commit e un riferimento a un ramo nel formato `refs/heads/branchname^{full-commit-SHA}`.
5. Selezionare Start build (Avvia compilazione).
6. Una volta completata la compilazione, si dovrebbe visualizzare quanto segue:
 - Nella scheda Build logs (Log di compilazione), la versione dell'origine del progetto che è stata utilizzata. Ecco un esempio.

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

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

- Nella scheda Environment variables (Variabili di ambiente), Resolved source version (Versione di origine risolta) corrisponde all'ID commit utilizzato per creare la compilazione.
- Nella scheda Phase details (Dettagli di fase), la durata della fase `DOWNLOAD_SOURCE` deve essere più breve di quella in cui si è utilizzato solo l'ID commit per specificare la versione dell'origine.

Esempi di repository di sorgenti di terze parti per CodeBuild

Questa sezione descrive esempi di integrazioni tra repository di sorgenti di terze parti e CodeBuild

Project N.E.M.O.	Description
BitBucket pull request e esempio di filtro webhook: vedi Esegui l'esempio «Bitbucket pull request and webhook filter» per CodeBuild	Questo esempio mostra come creare una richiesta pull tramite un repository Bitbucket . Mostra anche come utilizzare un webhook Bitbucket per attivare CodeBuild per creare una compilazione di un progetto.
GitHub Esempio di Enterprise Server: vedi Esegui l'esempio di GitHub Enterprise Server per CodeBuild	Questo esempio mostra come configurare i CodeBuild progetti quando nel repository di GitHub Enterprise Server è installato un certificato. Mostra anche come abilitare i webhook in modo da CodeBuild ricostruire il codice sorgente ogni volta che una modifica al codice viene inserita nell'archivio di Enterprise Server. GitHub
GitHub pull request e esempio di filtro webhook: vedi Esegui l'esempio di GitHub pull request e di filtro webhook per CodeBuild	Questo esempio mostra come creare una richiesta pull utilizzando un repository di GitHub Enterprise Server. Viene inoltre illustrato come abilitare i webhook in modo da CodeBuild ricostruire il codice sorgente ogni volta che una modifica al codice viene inserita nell'archivio di Enterprise Server. GitHub

Esegui l'esempio «Bitbucket pull request and webhook filter» per CodeBuild

AWS CodeBuild supporta i webhook quando il repository di origine è Bitbucket. Ciò significa che per un progetto di CodeBuild compilazione il cui codice sorgente è archiviato in un repository Bitbucket, i webhook possono essere utilizzati per ricostruire il codice sorgente ogni volta che viene inserita una modifica al codice nell'archivio. Per ulteriori informazioni, consulta [Eventi webhook Bitbucket](#).

Questo esempio mostra come creare una richiesta pull tramite un repository Bitbucket. Inoltre, mostra come utilizzare un webhook Bitbucket per attivare la creazione di una build di un progetto. CodeBuild

 Note

Quando si utilizzano i webhook, è possibile che un utente attivi una build inaspettata. Per mitigare questo rischio, consulta. [Procedure consigliate per l'utilizzo dei webhook](#)

Argomenti

- [Prerequisiti](#)
- [Passaggio 1: crea un progetto di compilazione con Bitbucket e abilita i webhook](#)
- [Passaggio 2: attiva una build con un webhook Bitbucket](#)

Prerequisiti

Per eseguire questo esempio devi connettere il tuo AWS CodeBuild progetto al tuo account Bitbucket.

 Note

CodeBuild ha aggiornato le sue autorizzazioni con Bitbucket. Se in precedenza hai collegato il tuo progetto a Bitbucket e ora ricevi un errore di connessione Bitbucket, devi riconnetterti per concedere l'autorizzazione a gestire i tuoi webhook. CodeBuild

Passaggio 1: crea un progetto di compilazione con Bitbucket e abilita i webhook

I passaggi seguenti descrivono come creare un AWS CodeBuild progetto con Bitbucket come repository di origine e abilitare i webhook.

1. [Apri la console su codebuild/home. AWS CodeBuild https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. Scegliere Create build project (Crea progetto di compilazione).
4. In Project configuration (Configuratore progetto):

Project name (Nome progetto)

Inserisci un nome per questo progetto di compilazione. I nomi dei progetti di build devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.

5. In Source (Origine):

Fornitore di origine

Scegli Bitbucket. Segui le istruzioni per connetterti (o riconnetterti) con Bitbucket, quindi scegli Autorizza.

Repository

Scegli Repository nel mio account Bitbucket.

Se non ti sei mai connesso in precedenza al tuo account Bitbucket, inserisci il nome utente e la password dell'app Bitbucket e seleziona Salva credenziali Bitbucket.

Repository Bitbucket

Inserisci l'URL del tuo repository Bitbucket.

6. In Primary source webhook events, seleziona quanto segue.

Note

La sezione Primary source webhook events è visibile solo se hai scelto Repository nel mio account Bitbucket nel passaggio precedente.

1. Al momento della creazione di un progetto, selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository).
2. Da Event type (Tipo di evento), selezionare uno o più eventi.
3. Per applicare un filtro che stabilisce quando un evento avvia una compilazione, in Start a build under these conditions (Avvia una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.

4. Per applicare un filtro che stabilisce quando un evento non avvia una compilazione, in Don't start a build under these conditions (Non avviare una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
5. Scegli Aggiungi gruppo di filtri per aggiungere un altro gruppo di filtri, se necessario.

Per ulteriori informazioni sui tipi di eventi e sui filtri del webhook di Bitbucket, consulta [Eventi webhook Bitbucket](#)

7. In Environment (Ambiente):

Immagine dell'ambiente

Seleziona una delle seguenti opzioni:

Per utilizzare un'immagine Docker gestita da AWS CodeBuild:

Scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Immagine e Versione dell'immagine. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).

Per usare un'altra immagine Docker:

Scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS

Per utilizzare un'immagine Docker privata:

Scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [What Is? Gestione dei segreti AWS](#) nella Guida Gestione dei segreti AWS per l'utente.

Ruolo di servizio

Seleziona una delle seguenti opzioni:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.

- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

 Note

Quando utilizzi la console per creare o aggiornare un progetto di build, puoi creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

8. In Buildspec, esegui una delle seguenti operazioni:

- Scegliete Usa un file buildspec per usare il file buildspec.yml nella directory principale del codice sorgente.
- Scegli Inserisci comandi di compilazione per utilizzare la console per inserire i comandi di compilazione.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

9. In Artifacts (Artefatti):

Tipo

Seleziona una delle seguenti opzioni:

- Se non si desidera creare artefatti di output di compilazione, scegliere No artifacts (Nessun artefatto).
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Per impostazione predefinita, il nome dell'artefatto è quello del progetto. Se si desidera utilizzare un nome diverso, immetterlo nella casella del nome degli artefatti. Se si desidera eseguire l'output di un file .ZIP, includere l'estensione .zip.
 - Per Bucket name (Nome bucket) selezionare il nome del bucket di output.

- Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, appspec.yml, target/my-app.jar). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).

Configurazione aggiuntiva

Espandere Additional configuration (Configurazione aggiuntiva) e impostare le opzioni come opportuno.

10. Scegliere Create build project (Crea progetto di compilazione). Nella pagina Review (Verifica), selezionare Start build (Avvia compilazione) per eseguire la compilazione.

Passaggio 2: attiva una build con un webhook Bitbucket

Per un progetto che utilizza webhook Bitbucket, AWS CodeBuild crea una build quando il repository Bitbucket rileva una modifica nel codice sorgente.

1. AWS CodeBuild <https://console.aws.amazon.com/codesuite/Apri> la console su codebuild/home.
2. Nel riquadro di navigazione, selezionare Build projects (Compilare progetti), quindi selezionare un progetto associato a un repository di Bitbucket con webhook. Per informazioni sulla creazione di un progetto webhook Bitbucket, consulta [the section called "Passaggio 1: crea un progetto di compilazione con Bitbucket e abilita i webhook"](#)
3. Apportare modifiche al codice nel repository Bitbucket del progetto.
4. Creare una richiesta pull nel repository Bitbucket. Per ulteriori informazioni, consultare l'articolo relativo alla [creazione di una richiesta pull](#).
5. Nella pagina dei webhook Bitbucket, selezionare View request (Visualizzare richiesta) per visualizzare un elenco di eventi recenti.
6. Scegli Visualizza dettagli per visualizzare i dettagli sulla risposta restituita da CodeBuild. Potrebbe essere simile al seguente:

```
"response": "Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
"statusCode": 200
```

7. Navigare alla pagina della richiesta pull Bitbucket per visualizzare lo stato della compilazione.

Esegui l'esempio di GitHub Enterprise Server per CodeBuild

AWS CodeBuild supporta GitHub Enterprise Server come repository di origine. Questo esempio mostra come configurare i CodeBuild progetti quando nel repository di GitHub Enterprise Server è installato un certificato. Mostra anche come abilitare i webhook in modo da CodeBuild ricostruire il codice sorgente ogni volta che una modifica al codice viene inserita nell'archivio di Enterprise Server. GitHub

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un progetto di compilazione con GitHub Enterprise Server e abilitare i webhook](#)

Prerequisiti

1. Genera un token di accesso personale per il tuo progetto. CodeBuild Ti consigliamo di creare un utente GitHub Enterprise e generare un token di accesso personale per questo utente. Copialo negli appunti in modo che possa essere utilizzato durante la CodeBuild creazione del progetto. Per ulteriori informazioni, consulta [Creazione di un token di accesso personale per la riga di comando sul sito Web](#) di GitHub Help.

Quando si crea il token di accesso personale, includere l'ambito del repo (repository) nella definizione.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

☒	repo	Full control of private repositories
☒	repo:status	Access commit status
☒	repo_deployment	Access deployment status
☒	public_repo	Access public repositories

2. Scarica il certificato da GitHub Enterprise Server. CodeBuild utilizza il certificato per stabilire una connessione SSL affidabile al repository.

Client Linux/macOS:

Dalla finestra di un terminale, eseguire il comando riportato qui sotto:

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```

Sostituire i segnaposto nel comando con i seguenti valori:

HOST. L'indirizzo IP del repository GitHub Enterprise Server.

PORTNUMBER. Il numero di porta che stai utilizzando per la connessione (ad esempio, 443).

folder. La cartella in cui hai scaricato il certificato.

filename. Il nome del file del certificato.

 Important

Salva il certificato come file .pem.

Client Windows:

Utilizzate il browser per scaricare il certificato da GitHub Enterprise Server. Per visualizzare i dettagli del certificato del sito, selezionare l'icona a forma di lucchetto. Per informazioni su come esportare il certificato, consultare la documentazione del browser.

 Important

Salva il certificato come file .pem.

3. Caricare il file di certificato in un bucket S3. Per informazioni su come creare un bucket S3, consulta [Come creare un bucket S3?](#) Per informazioni sul caricamento di oggetti in un bucket S3, consulta [Come caricare file e cartelle in un bucket.](#)

 Note

Questo bucket deve trovarsi nella stessa AWS regione delle tue build. Ad esempio, se si richiede di CodeBuild eseguire una build nella regione Stati Uniti orientali (Ohio), il bucket deve trovarsi nella regione Stati Uniti orientali (Ohio).

Fase 1: Creare un progetto di compilazione con GitHub Enterprise Server e abilitare i webhook

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
4. In Source, in Source provider, scegli GitHub Enterprise Server.
 - Scegli Gestisci le credenziali dell'account, quindi scegli Token di accesso personale. Per Service, scegli Secrets Manager (consigliato) e configura il tuo segreto. Quindi, in GitHub Enterprise Personal Access Token, inserisci il tuo token di accesso personale e scegli Salva.
 - In Repository URL (URL Repository), inserire il percorso del repository, compreso il nome del repository.
 - Espandere Additional configuration (Configurazione aggiuntiva).
 - Selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository) per rieseguire la compilazione ogni volta che viene inviata una modifica del codice a questo repository.
 - Seleziona Abilita SSL non sicuro per ignorare gli avvisi SSL durante la connessione al repository del progetto GitHub Enterprise Server.

Note

Si consiglia di utilizzare Enable insecure SSL (Abilita SSL non sicuro) solo nella fase di test. Non deve essere utilizzato in un ambiente di produzione.

Source

Add source

Source 1 - Primary

Source provider

GitHub Enterprise▼

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ Additional configuration

Git clone depth, Insecure SSL

Git clone depth - *optional*

1▼

Webhook - *optional*

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

Branch filter - *optional*

Enter a regular expression

Insecure SSL - *optional*

Enable this flag to ignore SSL warnings while connecting to project source.

☐ Enable insecure SSL

5. In Environment (Ambiente):

Per Environment image (Immagine ambiente), procedere in uno dei seguenti modi:

- Per utilizzare un'immagine Docker gestita da AWS CodeBuild, scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Image e Image version. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).
- Per utilizzare un'altra immagine Docker, selezionare Custom image (Immagine personalizzata). Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si

sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS

- Per utilizzare un'immagine Docker privata, scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

6. In Service role (Ruolo del servizio), procedere in uno dei seguenti modi:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.
- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

Note

Quando utilizzi la console per creare o aggiornare un progetto di build, puoi creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

7. Espandere Additional configuration (Configurazione aggiuntiva).

Se vuoi CodeBuild lavorare con il tuo VPC:

- Per VPC, scegli l'ID VPC che utilizza. CodeBuild
- Per le sottoreti VPC, scegli le sottoreti che includono le risorse che utilizza. CodeBuild
- Per i gruppi di sicurezza VPC, scegli i gruppi di sicurezza da CodeBuild utilizzare per consentire l'accesso alle risorse in. VPCs

Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

8. In Buildspec, esegui una delle seguenti operazioni:

- Scegliete Usa un file buildspec per usare il file buildspec.yml nella directory principale del codice sorgente.
- Scegli Inserisci comandi di compilazione per utilizzare la console per inserire i comandi di compilazione.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

9. In Artifacts (Artefatti), per Type (Tipo), procedere in uno dei seguenti modi:

- Se non si desidera creare artefatti di output di compilazione, scegliere No artifacts (Nessun artefatto).
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Per impostazione predefinita, il nome dell'artefatto è quello del progetto. Se si desidera utilizzare un nome diverso, immetterlo nella casella del nome degli artefatti. Se si desidera eseguire l'output di un file .ZIP, includere l'estensione .zip.
 - Per Bucket name (Nome bucket) selezionare il nome del bucket di output.
 - Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, appspec.yml, target/my-app.jar). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).

10. In Cache type (Tipo di cache), procedere in uno dei modi seguenti:

- Se non si desidera utilizzare una cache, scegliere No cache (Nessuna cache).
- Se desideri utilizzare una cache Amazon S3, scegli Amazon S3 e procedi come segue:
 - Per Bucket, selezionare il nome del bucket S3 in cui è archiviata la cache.
 - (Facoltativo) Per il prefisso del percorso della cache, inserisci un prefisso del percorso Amazon S3. Il valore Cache path prefix (Prefisso percorso cache) è simile a un nome di directory. Consente di archiviare la cache sotto la stessa directory in un bucket.

 Important

Non aggiungere una barra finale (/) alla fine del prefisso del percorso.

- Se si desidera utilizzare una cache locale, selezionare Local (Locale), quindi scegliere una o più modalità cache locali.

 Note

La modalità Cache di livello Docker è disponibile solo per Linux. Se la scegli, il progetto deve essere eseguito in modalità privilegiata.

L'utilizzo di una cache consente di risparmiare tempo di compilazione perché alcune parti riutilizzabili dell'ambiente di compilazione vengono memorizzate nella cache e utilizzate in compilazioni diverse. Per informazioni su come specificare una cache nel file di specifiche di compilazione, consulta [Sintassi buildspec](#). Per ulteriori informazioni sul caching, consulta [Creazioni di cache per migliorare le prestazioni](#).

11. Scegliere Create build project (Crea progetto di compilazione). Nella pagina di compilazione del progetto, selezionare Start build (Avvia compilazione).

Esegui l'esempio di GitHub pull request e di filtro webhook per CodeBuild

AWS CodeBuild supporta i webhook quando il repository di origine è GitHub. Ciò significa che per un progetto di CodeBuild compilazione il cui codice sorgente è archiviato in un GitHub repository, i webhook possono essere utilizzati per ricostruire il codice sorgente ogni volta che viene inserita una modifica al codice nel repository. [Per alcuni esempi, consulta Samples. CodeBuild AWS CodeBuild](#)

 Note

Quando si utilizzano i webhook, è possibile che un utente attivi una build inaspettata. Per mitigare questo rischio, vedi. [Procedure consigliate per l'utilizzo dei webhook](#)

Argomenti

- [Fase 1: Creare un progetto di compilazione con GitHub e abilitare i webhook](#)

- [Passaggio 2: verifica che i webhook siano abilitati](#)

Fase 1: Creare un progetto di compilazione con GitHub e abilitare i webhook

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. Scegliere Create build project (Crea progetto di compilazione).
4. In Project configuration (Configuratore progetto):

Project name (Nome progetto)

Inserisci un nome per questo progetto di compilazione. I nomi dei progetti di build devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.

5. In Source (Origine):

Fornitore di origine

Scegli GitHub. Segui le istruzioni per connetterti (o riconnetterti) con GitHub , quindi scegli Autorizza.

Repository

Scegli Repository nel mio account. GitHub

GitHub deposito

Inserisci l'URL del tuo GitHub repository.

6. In Primary source webhook events, seleziona quanto segue.

Note

La sezione Primary source webhook events è visibile solo se hai scelto Repository nel mio GitHub account nel passaggio precedente.

1. Al momento della creazione di un progetto, selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository).
2. Da Event type (Tipo di evento), selezionare uno o più eventi.
3. Per applicare un filtro che stabilisce quando un evento avvia una compilazione, in Start a build under these conditions (Avvia una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
4. Per applicare un filtro che stabilisce quando un evento non avvia una compilazione, in Don't start a build under these conditions (Non avviare una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
5. Scegli Aggiungi gruppo di filtri per aggiungere un altro gruppo di filtri, se necessario.

Per ulteriori informazioni sui tipi di eventi e sui filtri dei GitHub webhook, consulta [GitHub eventi webhook](#).

7. In Environment (Ambiente):

Immagine dell'ambiente

Seleziona una delle seguenti opzioni:

Per utilizzare un'immagine Docker gestita da AWS CodeBuild:

Scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Immagine e Versione dell'immagine. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).

Per usare un'altra immagine Docker:

Scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS

Per utilizzare un'immagine Docker privata:

Scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [What Is? Gestione dei segreti AWS](#) nella Guida Gestione dei segreti AWS per l'utente.

Ruolo di servizio

Seleziona una delle seguenti opzioni:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.
- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

Note

Quando utilizzi la console per creare o aggiornare un progetto di build, puoi creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

8. In Buildspec, esegui una delle seguenti operazioni:

- Scegliete Usa un file buildspec per usare il file buildspec.yml nella directory principale del codice sorgente.
- Scegli Inserisci comandi di compilazione per utilizzare la console per inserire i comandi di compilazione.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

9. In Artifacts (Artefatti):

Tipo

Seleziona una delle seguenti opzioni:

- Se non si desidera creare artefatti di output di compilazione, scegliere No artifacts (Nessun artefatto).
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Per impostazione predefinita, il nome dell'artefatto è quello del progetto. Se si desidera utilizzare un nome diverso, immetterlo nella casella del nome degli artefatti. Se si desidera eseguire l'output di un file .ZIP, includere l'estensione .zip.
- Per Bucket name (Nome bucket) selezionare il nome del bucket di output.
- Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, appspec.yml, target/my-app.jar). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).

Configurazione aggiuntiva

Espandere Additional configuration (Configurazione aggiuntiva) e impostare le opzioni come opportuno.

10. Scegliere Create build project (Crea progetto di compilazione). Nella pagina Review (Verifica), selezionare Start build (Avvia compilazione) per eseguire la compilazione.

Passaggio 2: verifica che i webhook siano abilitati

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Esegui una delle seguenti operazioni:
 - Selezionare il collegamento del progetto di compilazione con i webhook da verificare, quindi selezionare Build details (Dettagli compilazione).
 - Scegli il pulsante accanto al progetto di compilazione con i webhook che desideri verificare, scegli Visualizza dettagli, quindi scegli la scheda Dettagli della build.
4. In Primary source webhook events, scegli il link URL Webhook.

5. Nel tuo GitHub repository, nella pagina Impostazioni, sotto Webhook, verifica che siano selezionate le Pull Requests e Push.
6. Nelle impostazioni del tuo GitHub profilo, in Impostazioni personali, Applicazioni, OAuthApp autorizzate, dovresti vedere che l'applicazione è stata autorizzata ad accedere alla AWS regione selezionata.

Tutorial: firma del codice Apple con Fastlane nell' CodeBuild utilizzo di S3 per l'archiviazione dei certificati

[fastlane](#) è un popolare strumento di automazione open source per automatizzare le distribuzioni e le versioni beta per le tue app iOS e Android. Gestisce tutte le attività noiose, come la generazione di schermate, la firma del codice e il rilascio dell'applicazione.

Prerequisiti

Per completare questo tutorial, devi prima aver impostato quanto segue:

- Un Account AWS
- Un [account Apple Developer](#)
- Un bucket S3 per l'archiviazione dei certificati
- fastlane installato nel tuo progetto - [Guida](#) all'installazione di fastlane

Passaggio 1: configura Fastlane Match con S3 sul tuo computer locale

[Fastlane Match](#) è uno degli [strumenti di Fastlane](#) e consente una configurazione senza interruzioni per la firma del codice sia nell'ambiente di sviluppo locale che in quello successivo. CodeBuild Fastlane Match archivia tutti i certificati di firma del codice e i profili di provisioning in un repository/S3 Bucket/Google cloud storage Git e scarica e installa i certificati e i profili necessari quando richiesto.

In questa configurazione di esempio, configurerai e utilizzerai un bucket Amazon S3 per lo storage.

1. Inizializza match nel tuo progetto:

```
fastlane match init
```

2. Quando richiesto, scegli S3 come modalità di archiviazione.

3. Aggiorna il tuo `Matchfile` per usare S3:

```
storage_mode("s3")
  s3_bucket("your-s3-bucket-name")
  s3_region("your-aws-region")
  type("appstore") # The default type, can be: appstore, adhoc, enterprise or
development
```

Passaggio 2: configura il tuo Fastfile

Crea o aggiorna il tuo `Fastfile` con la seguente corsia.

Sì CodeBuild, Fastlane Match dovrà essere eseguito ogni volta che crei e firmi la tua app. Il modo più semplice per farlo è aggiungere l'azione alla corsia in cui viene creata l'app.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

Note

Assicurati di aggiungerla `setup_ci` alla `before_all` sezione in `Fastfile` affinché l'azione della partita funzioni correttamente. Ciò garantisce che venga utilizzato un portachiavi Fastlane temporaneo con le autorizzazioni appropriate. Senza utilizzarlo, è possibile che si verifichino errori di compilazione o risultati incoerenti.

Passaggio 3: Esegui il **fastlane match** comando per generare i rispettivi certificati e profili

Il comando fastlane match per il tipo specificato (ad esempio development, appstore, adhoc, enterprise) genererà il certificato e il profilo se non sono disponibili nell'archivio remoto. I certificati e i profili verranno archiviati in S3 da fastlane.

```
bundle exec fastlane match appstore
```

L'esecuzione del comando sarà interattiva e fastlane chiederà di impostare la passphrase per decrittografare i certificati.

Fase 4: Crea il file dell'applicazione per il tuo progetto

Create o aggiungete il file dell'applicazione appropriato per il vostro progetto.

1. Crea o aggiungi [Gymfile](#), [Appfile](#), [Snapfile](#), [Deliverfile](#) in base ai requisiti di costruzione del tuo progetto.
2. Effettua le modifiche nel tuo repository remoto

Fase 5: Creare variabili di ambiente in Secrets Manager

Crea due segreti per memorizzare il cookie di sessione fastlane e la passphrase corrispondente. Per ulteriori informazioni sulla creazione di segreti in Secrets Manager, consulta [Creare un Gestione dei segreti AWS segreto](#).

1. Accedi al cookie di sessione fastlane come segue.
 - a. Chiave segreta - FASTLANE_SESSION
 - b. Valore segreto: cookie di sessione generato dall'esecuzione del seguente comando sul computer locale.

Note

Questo valore è disponibile dopo l'autenticazione in un file locale: `~/fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <apple account>
```

2. Password Fastlane Match - Per consentire a Fastlane Match di decrittografare i certificati e i profili archiviati nel bucket S3, è necessario aggiungere la passphrase di crittografia configurata nella fase di configurazione di Match alle variabili di ambiente del progetto. CodeBuild
 - a. Chiave segreta - MATCH_PASSWORD
 - b. Valore segreto - *<match passphrase to decrypt certificates>*. La passphrase viene impostata durante la generazione dei certificati nel passaggio 3.

Note

Durante la creazione dei segreti di cui sopra in Secrets Manager, ricordati di assegnare un nome segreto con il seguente prefisso: /CodeBuild/

Fase 6: Creare una flotta di elaborazione

Crea la flotta di elaborazione per il tuo progetto.

1. Nella console, vai a CodeBuild e crea una nuova flotta di elaborazione.
2. Scegli «macOS» come sistema operativo e seleziona un tipo di calcolo e un'immagine appropriati.

Passaggio 7: Crea un progetto in CodeBuild

Crea il tuo progetto in CodeBuild.

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
3. Configura il tuo provider di origine (ad esempio,). GitHub CodeCommit Questo è il repository dei sorgenti del progetto iOS e non l'archivio dei certificati.

4. In Environment (Ambiente):

- Scegli Capacità riservata.
- Per Fleet, seleziona la flotta creata sopra.
- Fornisci il nome del ruolo di servizio che CodeBuild creerai per te.
- Fornisci le seguenti variabili di ambiente.
 - Nome: MATCH_PASSWORD, Valore: *<secrets arn>*, Tipo: Secrets Manager (Secrets ARN creato nel passaggio 5 per MATCH_PASSWORD)
 - Nome: FASTLANE_SESSION, Valore: *<secrets arn>*, Tipo: Secrets Manager (Secrets ARN creato nel passaggio 5 per FASTLANE_SESSION)

5. In Buildspec, aggiungi quanto segue:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Fase 8: Configurazione del ruolo IAM

Una volta creato il progetto, assicurati che il ruolo di servizio del CodeBuild progetto disponga delle autorizzazioni per accedere al bucket S3 contenente i certificati. Aggiungi la seguente politica al ruolo:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::your-s3-bucket-name"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
    }
  ]
}
```

Passaggio 9: Esegui la build

Esegui la compilazione. Puoi controllare lo stato della build e accedere. CodeBuild

Una volta completato il lavoro, potrai visualizzare il registro del lavoro.

Risoluzione dei problemi

- Se riscontri problemi con il recupero dei certificati, assicurati che le autorizzazioni IAM siano configurate correttamente per l'accesso a S3.
- Se riscontri problemi con la decrittografia dei certificati, assicurati di impostare la passphrase corretta nella variabile di ambiente MATCH_PASSWORD.
- Per problemi di firma del codice, verifica che il tuo account Apple Developer disponga dei certificati e dei profili necessari e che l'identificatore del pacchetto nel tuo progetto Xcode corrisponda a quello nel tuo profilo di provisioning.

Considerazioni relative alla sicurezza

Di seguito sono riportate le considerazioni sulla sicurezza relative a questo tutorial.

- Assicurati che il tuo bucket S3 abbia le impostazioni di sicurezza appropriate, inclusa la crittografia a riposo. In particolare, assicurati che il bucket non abbia accesso pubblico e limita l'accesso solo CodeBuild al sistema che deve avere accesso.
- Prendi in considerazione l'utilizzo Gestione dei segreti AWS per archiviare informazioni sensibili come MATCH_PASSWORD e FASTLANE_SESSION.

Questo esempio fornisce una configurazione per la firma del codice iOS con Fastlane CodeBuild utilizzando Amazon S3 per l'archiviazione dei certificati. Potrebbe essere necessario modificare alcuni passaggi in base ai requisiti e CodeBuild all'ambiente specifici del progetto. Questo approccio sfrutta i AWS servizi per una maggiore sicurezza e integrazione all'interno dell' AWS ecosistema.

Tutorial: utilizzo della firma del codice Apple con Fastlane GitHub per CodeBuild l'archiviazione dei certificati

[fastlane](#) è un popolare strumento di automazione open source per automatizzare le distribuzioni e le versioni beta per le tue app iOS e Android. Gestisce tutte le attività noiose, come la generazione di schermate, la firma del codice e il rilascio dell'applicazione.

Questo esempio dimostra come configurare la firma del codice Apple utilizzando Fastlane in un CodeBuild progetto in esecuzione su Mac Fleet, GitHub come spazio di archiviazione per certificati e profili di provisioning.

Prerequisiti

Per completare questo tutorial, devi prima aver impostato quanto segue:

- Un Account AWS
- Un [account Apple Developer](#)
- Un GitHub archivio privato per l'archiviazione dei certificati
- fastlane installato nel tuo progetto - [Guida](#) all'installazione di fastlane

Passaggio 1: configura Fastlane Match con GitHub sul tuo computer locale

[Fastlane Match](#) è uno degli [strumenti di Fastlane](#) e consente una configurazione senza interruzioni per la firma del codice sia nell'ambiente di sviluppo locale che in quello successivo. CodeBuild Fastlane Match archivia tutti i certificati di firma del codice e i profili di provisioning in un repository/S3 Bucket/Google cloud storage Git e scarica e installa i certificati e i profili necessari quando richiesto.

In questa configurazione di esempio, configureremo e utilizzeremo un repository Git per l'archiviazione.

1. Inizializza match nel tuo progetto:

```
fastlane match init
```

2. Quando richiesto, scegli GitHub come modalità di archiviazione.
3. Aggiorna il tuo `Matchfile` per usare: GitHub

```
git_url("https://github.com/your-username/your-certificate-repo.git")
storage_mode("git")
type("development") # The default type, can be: appstore, adhoc, enterprise or
development
```

Note

Assicurati di inserire l'URL HTTPS per il tuo repository Git per l'autenticazione e la clonazione con successo di fastlane. Altrimenti, potresti visualizzare un errore di autenticazione quando tenti di utilizzare match.

Passaggio 2: configura il tuo Fastfile

Crea o aggiorna il tuo `Fastfile` con la seguente corsia.

Sì CodeBuild, Fastlane Match dovrà essere eseguito ogni volta che crei e firmi la tua app. Il modo più semplice per farlo è aggiungere l'archiviazione alla corsia in cui viene creata l'app.

```
default_platform(:ios)
```

```
platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

Note

Assicurati di aggiungerla `setup_ci` alla `before_all` sezione in `Fastfile` affinché l'azione della partita funzioni correttamente. Ciò garantisce che venga utilizzato un portachiavi Fastlane temporaneo con le autorizzazioni appropriate. Senza utilizzarlo, è possibile che si verifichino errori di compilazione o risultati incoerenti.

Passaggio 3: Esegui il **fastlane match** comando per generare i rispettivi certificati e profili

Il comando `fastlane match` per il tipo specificato (ad esempio `development`, `appstore`, `adhoc`, `enterprise`) genererà il certificato e il profilo se non sono disponibili nell'archivio remoto. I certificati e i profili verranno archiviati da fastlane. GitHub

```
bundle exec fastlane match appstore
```

L'esecuzione del comando sarà interattiva e fastlane chiederà di impostare la passphrase per decrittografare i certificati.

Fase 4: Crea il file dell'applicazione per il tuo progetto

Create o aggiungete il file dell'applicazione appropriato per il vostro progetto.

1. Crea o aggiungi [Gymfile](#), [Appfile](#), [Snapfile](#), [Deliverfile](#) in base ai requisiti di costruzione del tuo progetto.
2. Effettua le modifiche nel tuo repository remoto.

Fase 5: Creare variabili di ambiente in Secrets Manager

Crea tre segreti per memorizzare il cookie di sessione fastlane e la passphrase corrispondente. Per ulteriori informazioni sulla creazione di segreti in Secrets Manager, consulta [Creare un Gestione dei segreti AWS segreto](#).

1. Accedi al cookie di sessione fastlane come segue.
 - a. Chiave segreta - FASTLANE_SESSION
 - b. Valore segreto: cookie di sessione generato dall'esecuzione del seguente comando sul computer locale.

Note

Questo valore è disponibile dopo l'autenticazione in un file locale: `~/fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <Apple_account>
```

2. Password Fastlane Match - Per consentire a Fastlane Match di decrittografare i certificati e i profili archiviati nel repository Git, è necessario aggiungere la passphrase di crittografia configurata nella fase di configurazione di Match alle variabili di ambiente del progetto.
CodeBuild
 - a. Chiave segreta - MATCH_PASSWORD
 - b. Valore segreto - `<match passphrase to decrypt certificates>`. La passphrase viene impostata durante la generazione dei certificati nel passaggio 3.
3. `FastlaneMATCH_GIT_BASIC_AUTHORIZATION`: imposta un'autorizzazione di base per la partita:
 - a. Chiave segreta:

MATCH_GIT_BASIC_AUTHORIZATION

- b. Valore segreto: il valore deve essere una stringa codificata in base 64 contenente il nome utente e il token di accesso personale (PAT) nel formato. `username:password` Puoi generarlo usando il seguente comando:

```
echo -n your_github_username:your_personal_access_token | base64
```

Puoi generare il tuo PAT sulla GitHub console in Il mio profilo > Impostazioni > Impostazioni degli sviluppatori > Token di accesso personale. [Per ulteriori informazioni, consulta la seguente guida: -. https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens)

Note

Durante la creazione dei segreti di cui sopra in Secrets Manager, ricordati di assegnare un nome segreto con il seguente prefisso: `/CodeBuild/`

Fase 6: Creare una flotta di elaborazione

Crea la flotta di elaborazione per il tuo progetto.

1. Nella console, vai a CodeBuild e crea una nuova flotta di elaborazione.
2. Scegli macOS come sistema operativo e seleziona il tipo di elaborazione e l'immagine appropriati.

Fase 7: Creare un progetto in CodeBuild

Crea il tuo progetto in CodeBuild.

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
3. Configura il tuo provider di origine (ad esempio,). GitHub CodeCommit Questo è il repository dei sorgenti del progetto iOS e non l'archivio dei certificati.

4. In Environment (Ambiente):

- Scegli Capacità riservata.
- Per Fleet, seleziona la flotta creata sopra.
- Fornisci il nome del ruolo di servizio che CodeBuild creerà per te.
- Fornisci le seguenti variabili di ambiente.
 - Nome: MATCH_PASSWORD, Valore: *<secrets arn>*, Tipo: Secrets Manager (Secrets ARN creato nel passaggio 5 per MATCH_PASSWORD)
 - Nome: FASTLANE_SESSION, Valore: *<secrets arn>*, Tipo: Secrets Manager (Secrets ARN creato nel passaggio 5 per FASTLANE_SESSION)
 - Nome: MATCH_GIT_BASIC_AUTHORIZATION, Valore: *<secrets ARN>*, Tipo: Secrets Manager Secrets ARN (creato nel passaggio 5 per) MATCH_GIT_BASIC_AUTHORIZATION

5. In Buildspec, aggiungi quanto segue:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Passaggio 8: esegui la build

Eseguire la compilazione. Puoi controllare lo stato della build e accedere. CodeBuild

Una volta completato il lavoro, potrai visualizzare il registro del lavoro.

Risoluzione dei problemi

- Se riscontri problemi di accesso al GitHub repository, ricontrolla il tuo token di accesso personale e la variabile di ambiente `MATCH_GIT_BASIC_AUTHORIZATION`.
- Se riscontri problemi con la decrittografia dei certificati, assicurati di impostare la passphrase corretta nella variabile di ambiente `MATCH_PASSWORD`.
- Per problemi di firma del codice, verifica che il tuo account Apple Developer disponga dei certificati e dei profili necessari e che l'identificatore del pacchetto nel tuo progetto Xcode corrisponda a quello nel tuo profilo di provisioning.

Considerazioni relative alla sicurezza

Di seguito sono riportate le considerazioni sulla sicurezza relative a questo tutorial.

- Mantieni privato il tuo GitHub archivio di certificati e verifica regolarmente l'accesso.
- Prendi in considerazione l'utilizzo Gestione dei segreti AWS per l'archiviazione di informazioni sensibili come `MATCH_PASSWORD` e `FASTLANE_SESSION`.

Questo esempio fornisce una configurazione per la firma del codice iOS con Fastlane da CodeBuild utilizzare GitHub per l'archiviazione dei certificati. Potrebbe essere necessario modificare alcuni passaggi in base ai requisiti e all' CodeBuild ambiente specifici del progetto. Questo approccio sfrutta i AWS servizi per una maggiore sicurezza e integrazione all'interno dell' AWS ecosistema.

Imposta i nomi degli artefatti in fase di compilazione utilizzando il controllo delle versioni semantiche

Questo campione contiene esempi di file di specifiche di compilazione che spiegano come specificare il nome di un artefatto che viene creato durante il processo di compilazione. Il nome specificato nel file di specifiche di compilazione può includere comandi Shell e variabili di ambiente al fine di renderlo univoco. Il nome specificato nel file di specifiche di compilazione sostituirà il nome immesso nella console al momento della creazione del progetto.

Se si ripete il processo di compilazione più volte, l'utilizzo di un nome dell'artefatto specificato nel file di specifiche di compilazione farà in modo che i nomi dei file artefatti di output siano univoci.

Ad esempio, è possibile utilizzare una data e un timestamp che sono inseriti all'interno del nome dell'artefatto durante il processo di compilazione.

Per sostituire il nome dell'artefatto che si è inserito nella console con il nome presente nel file di specifiche di compilazione, basta seguire questi passaggi:

1. Impostare il progetto di compilazione per sostituire il nome dell'artefatto con il nome presente nel file di specifiche di compilazione.
 - Se utilizzi la console per creare il progetto di compilazione, seleziona **Enable semantic versioning** (Abilita funzione Versioni multiple semantica). Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#).
 - Se usi il AWS CLI, imposta su `overrideArtifactName true` nel file in formato JSON passato a `create-project`. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).
 - Se utilizzi l' AWS CodeBuild API, imposta il `overrideArtifactName` flag sull'`ProjectArtifacts` soggetto quando viene creato o aggiornato un progetto o viene avviata una build.
2. Specificare il nome nel file di specifiche di compilazione. Utilizza il seguente esempio di file di specifiche di compilazione come guida.

Questo esempio di Linux mostra come specificare il nome di un artefatto che comprende la data di creazione della compilazione.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${date +%Y-%m-%d}
```

Questo esempio di Linux mostra come specificare il nome di un artefatto che utilizza una variabile di ambiente CodeBuild. Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).

```
version: 0.2
```

```
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

Questo esempio di Windows mostra come specificare il nome di un artefatto che comprende data e ora di creazione della compilazione.

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Questo esempio di Windows mostra come specificare il nome di un artefatto che utilizza una variabile dichiarata nel file buildspec e una variabile di ambiente. CodeBuild Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
```



```
files:  
  - '**/*'  
name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Per ulteriori informazioni, consulta [Riferimento alle specifiche di costruzione per CodeBuild](#).

Esegui esempi di Microsoft Windows per CodeBuild

Questi esempi utilizzano un ambiente di AWS CodeBuild compilazione che esegue Microsoft Windows Server 2019, .NET Framework e .NET Core SDK per creare file di runtime a partire da codice scritto in F# e Visual Basic.

Important

L'esecuzione di questi esempi potrebbe comportare addebiti sul tuo AWS account. Questi includono eventuali addebiti per CodeBuild e per AWS risorse e azioni relative ad Amazon S3 e AWS KMS CloudWatch Logs. Per ulteriori informazioni, consulta [CodeBuild prezzi, i prezzi](#) di [Amazon S3, i prezzi](#) e [AWS Key Management Service i prezzi](#) di [Amazon CloudWatch](#).

Esegui gli esempi di Windows

Utilizzare la procedura seguente per eseguire gli esempi di Windows.

Per eseguire gli esempi di Windows

1. Crea i file come descritto nelle [File](#) sezioni [Struttura delle directory](#) e di questo argomento, quindi caricali in un bucket di input S3 o in un repository CodeCommit or GitHub.

Important

Non caricare *(root directory name)*, ma solo i file all'interno di *(root directory name)*

Se si sta utilizzando un bucket di input S3, assicurarsi di creare un file ZIP contenente i file, quindi caricarlo nel bucket di input. Non aggiungere *(root directory name)* al file ZIP, ma solo i file all'interno di *(root directory name)*

2. Creare un progetto di compilazione. Il progetto di compilazione deve utilizzare `l'mcr.microsoft.com/dotnet/framework/sdk:4.8` immagine per creare progetti .NET Framework.

Se si utilizza il AWS CLI per creare il progetto di compilazione, l'input in formato JSON del `create-project` comando potrebbe avere un aspetto simile a questo. (Sostituire i segnaposto con i propri valori).

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Esegui la build e segui i passaggi indicati. [Esegui le build manualmente](#)
4. Per ottenere l'artefatto di output della compilazione nel bucket di output S3 scaricare il file *windows-build-output-artifact.zip* nel computer o nell'istanza locale. Estrai il contenuto per accedere al runtime e ad altri file.
 - Il file di runtime per l'esempio F# che utilizza .NET FrameworkFSharpHelloWorld.exe, si trova nella FSharpHelloWorld\bin\Debug directory.
 - Il file di runtime per l'esempio di Visual Basic che utilizza .NET FrameworkVBHelloWorld.exe,, si trova nella VBHelloWorld\bin\Debug directory.

Struttura delle directory

Tali esempi assumono le seguenti strutture di directory.

F# e .NET Framework

(root directory name)

```
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
### App.config
### AssemblyInfo.fs
### FSharpHelloWorld.fsproj
### Program.fs
```

Visual Basic e .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
### App.config
### HelloWorld.vb
### VBHelloWorld.vbproj
### My Project
### Application.Designer.vb
### Application.myapp
### AssemblyInfo.vb
### Resources.Designer.vb
### Resources.resx
### Settings.Designer.vb
### Settings.settings
```

File

Questi esempi utilizzano i file seguenti.

F# e .NET Framework

buildspec.yml (in *(root directory name)*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8
```

```

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
        Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
  artifacts:
    files:
      - .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln (in *(root directory name)*):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config (in *(root directory name)\FSharpHelloWorld*):

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
```

```
</configuration>
```

AssemblyInfo.fs (in *(root directory name)*\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]
```

```
do
  ()
```

FSharpHelloWorld.fsproj (in *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <Tailcalls>>false</Tailcalls>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <Tailcalls>true</Tailcalls>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
```

```

    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="mscorlib" />
    <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
      <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Numerics" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="AssemblyInfo.fs" />
    <Compile Include="Program.fs" />
    <None Include="App.config" />
  </ItemGroup>
  <PropertyGroup>
    <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''">11</
MinimumVisualStudioVersion>
  </PropertyGroup>
  <Choose>
    <When Condition="'$(VisualStudioVersion)' == '11.0'">
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </When>
    <Otherwise>
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </Otherwise>
  </Choose>
  <Import Project="$(FSharpTargetsPath)" />
  <!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.

```



```
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>
```

Program.fs (in *(root directory name)*\FSharpHelloWorld):

```
// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code
```

Visual Basic e .NET Framework

buildspec.yml (in *(root directory name)*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
  artifacts:
    files:
      - .\VBHelloWorld\bin\Debug\*
```

VBHelloWorld.sln (in *(root directory name)*):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal

```

App.config (in *(root directory name)\VBHelloWorld*):

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
    </startup>
</configuration>

```

HelloWorld.vb (in *(root directory name)\VBHelloWorld*):

```

Module HelloWorld

    Sub Main()
        MsgBox("Hello World")
    End Sub

End Module

```

VBHelloWorld.vbproj (in *(root directory name)*\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
```

```
</PropertyGroup>
<PropertyGroup>
  <OptionCompare>Binary</OptionCompare>
</PropertyGroup>
<PropertyGroup>
  <OptionStrict>Off</OptionStrict>
</PropertyGroup>
<PropertyGroup>
  <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Xml" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
  <Import Include="Microsoft.VisualBasic" />
  <Import Include="System" />
  <Import Include="System.Collections" />
  <Import Include="System.Collections.Generic" />
  <Import Include="System.Data" />
  <Import Include="System.Diagnostics" />
  <Import Include="System.Linq" />
  <Import Include="System.Xml.Linq" />
  <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
  <Compile Include="HelloWorld.vb" />
  <Compile Include="My Project\AssemblyInfo.vb" />
  <Compile Include="My Project\Application.Designer.vb">
    <AutoGen>True</AutoGen>
    <DependentUpon>Application.myapp</DependentUpon>
  </Compile>
  <Compile Include="My Project\Resources.Designer.vb">
    <AutoGen>True</AutoGen>
    <DesignTime>True</DesignTime>
    <DependentUpon>Resources.resx</DependentUpon>
  </Compile>
  <Compile Include="My Project\Settings.Designer.vb">
```

```

    <AutoGen>True</AutoGen>
    <DependentUpon>Settings.settings</DependentUpon>
    <DesignTimeSharedInput>True</DesignTimeSharedInput>
  </Compile>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="My Project\Resources.resx">
    <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.vb</LastGenOutput>
    <CustomToolNamespace>My.Resources</CustomToolNamespace>
    <SubType>Designer</SubType>
  </EmbeddedResource>
</ItemGroup>
<ItemGroup>
  <None Include="My Project\Application.myapp">
    <Generator>MyApplicationCodeGenerator</Generator>
    <LastGenOutput>Application.Designer.vb</LastGenOutput>
  </None>
  <None Include="My Project\Settings.settings">
    <Generator>SettingsSingleFileGenerator</Generator>
    <CustomToolNamespace>My</CustomToolNamespace>
    <LastGenOutput>Settings.Designer.vb</LastGenOutput>
  </None>
  <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Application.Designer.vb (in *(root directory name)\VBHelloWorld\My Project*):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'

```

```
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.
' </auto-generated>
' -----
```

```
Option Strict On
Option Explicit On
```

Application.myapp (in *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb (in *(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>
```

```
'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

Resources.Designer.vb (in *(root directory name)*\VBHelloWorld\My Project):

```
'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On

Namespace My.Resources

    'This class was auto-generated by the StronglyTypedResourceBuilder
    'class via a tool like ResGen or Visual Studio.
    'To add or remove a member, edit your .ResX file then rerun ResGen
    'with the /str option, or rebuild your VS project.
    '''<summary>
    '''   A strongly-typed resource class, for looking up localized strings, etc.
    '''</summary>
```

```

<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
    Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
    Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
    Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
    Friend Module Resources

        Private resourceMan As Global.System.Resources.ResourceManager

        Private resourceCulture As Global.System.Globalization.CultureInfo

        '''<summary>
        ''' Returns the cached ResourceManager instance used by this class.
        '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
    Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
    Get
        If Object.ReferenceEquals(resourceMan, Nothing) Then
            Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
            resourceMan = temp
        End If
        Return resourceMan
    End Get
End Property

    '''<summary>
    ''' Overrides the current thread's CurrentUICulture property for all
    ''' resource lookups using this strongly typed resource class.
    '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
    Friend Property Culture() As Global.System.Globalization.CultureInfo
    Get
        Return resourceCulture
    End Get
    Set(ByVal value As Global.System.Globalization.CultureInfo)
        resourceCulture = value

```



```

    End Set
  End Property
End Module
End Namespace

```

Resources.resx (in *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
```

```
<!--
```

```
  Microsoft ResX Schema
```

```
  Version 2.0
```

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

```
... ado.net/XML headers & schema ...
```

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
  <value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```

```
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
```

```
  <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
```

```
  <comment>This is a comment</comment>
```

```
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

```
mimetype: application/x-microsoft.net.object.binary.base64
value    : The object must be serialized with
          : System.Serialization.Formatters.Binary.BinaryFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.soap.base64
value    : The object must be serialized with
          : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.bytearray.base64
value    : The object must be serialized into a byte array
          : using a System.ComponentModel.TypeConverter
          : and then encoded with base64 encoding.
```

```
-->
```

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="metadata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string" />
            <xsd:attribute name="type" type="xsd:string" />
            <xsd:attribute name="mimetype" type="xsd:string" />
          </xsd:complexType>
```

```

    </xsd:element>
    <xsd:element name="assembly">
      <xsd:complexType>
        <xsd:attribute name="alias" type="xsd:string" />
        <xsd:attribute name="name" type="xsd:string" />
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="data">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
          <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
        <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
        <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="resheader">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required" />
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
  <value>text/microsoft-resx</value>
</resheader>
<resheader name="version">
  <value>2.0</value>
</resheader>
<resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">

```

```

    <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
</root>

```

Settings.Designer.vb (in *(root directory name)\VBHelloWorld\My Project*):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

```

```

Option Strict On
Option Explicit On

```

```

Namespace My

```

```

    <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _

```

```

    Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
    "11.0.0.0"), _

```

```

    Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
    _

```

```

    Partial Friend NotInheritable Class MySettings

```

```

        Inherits Global.System.Configuration.ApplicationSettingsBase

```

```

        Private Shared defaultInstance As MySettings =
        CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
        MySettings), MySettings)

```

```

        #Region "My.Settings Auto-Save Functionality"

```

```

            #If _MyType = "WindowsForms" Then

```

```

                Private Shared addedHandler As Boolean

```

```

                Private Shared addedHandlerLockObject As New Object

```

```

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
        Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
            If My.Application.SaveMySettingsOnExit Then
                My.Settings.Save()
            End If
        End Sub
    #End If
#End Region

Public Shared ReadOnly Property [Default]() As MySettings
    Get

        #If _MyType = "WindowsForms" Then
            If Not addedHandler Then
                SyncLock addedHandlerLockObject
                    If Not addedHandler Then
                        AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                        addedHandler = True
                    End If
                End SyncLock
            End If
        #End If
        Return defaultInstance
    End Get
End Property
End Class
End Namespace

Namespace My

    <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute()> _
    Friend Module MySettingsProperty

        <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
        Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
            Get
                Return Global.VBHelloWorld.My.MySettings.Default
            End Get
        End Property
    
```

```
End Module
End Namespace
```

Settings.settings (in *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
  CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

Pianifica un build in AWS CodeBuild

Prima di utilizzarlo AWS CodeBuild, è necessario rispondere a queste domande:

1. Dove è memorizzato il codice sorgente? CodeBuild attualmente supporta la compilazione dai seguenti fornitori di repository di codice sorgente. Il codice sorgente deve contenere un file di specifica di compilazione (buildspec). Un buildspec è una raccolta di comandi di compilazione e impostazioni correlate, in formato YAML, che CodeBuild viene utilizzata per eseguire una build. È possibile dichiarare un buildspec nella definizione di un progetto di compilazione.

Provider di repository	Richiesto	Documentazione
CodeCommit	Nome del repository. (Facoltativo) ID del commit associato al codice sorgente.	Consulta questi argomenti nella Guida per l'utente di AWS CodeCommit : Crea un repository CodeCommit Crea un commit in CodeCommit
Amazon S3	Nome del bucket in entrata. Nome dell'oggetto corrispondente al file ZIP della compilazione in entrata che contiene il codice sorgente. (Facoltativo) ID della versione associata al file ZIP della	Consulta questi argomenti nella Guida introduttiva di Amazon S3: Creazione di un bucket Aggiunta di un oggetto a un bucket.

Provider di repository	Richiesto	Documentazione
	compilazione in entrata.	
GitHub	Nome del repository. (Facoltativo) ID del commit associato al codice sorgente.	Consulta questo argomento sul sito Web di GitHub assistenza: Creazione di un repository
Bitbucket	Nome del repository. (Facoltativo) ID del commit associato al codice sorgente.	Consulta questo argomento sul sito Web relativo alla documentazione di Bitbucket Cloud: Sezione relativa alla creazione di un repository

2. Quali comandi di compilazione devi eseguire e in quale ordine? Per impostazione predefinita, CodeBuild scarica l'input della build dal provider specificato e carica l'output della build nel bucket specificato. Utilizzare la specifica di compilazione per indicare in che modo si desidera convertire l'input di compilazione scaricato nell'output di compilazione previsto. Per ulteriori informazioni, consulta [Riferimento per buildspec](#).
3. Quali runtime e quali strumenti sono necessari per eseguire la compilazione? Ad esempio, stai compilando per Java, Ruby, Python, o Node.js? Hai bisogno di Maven, Ant o un compilatore per Java, Ruby o Python per la compilazione? La build richiede Git AWS CLI, the o altri strumenti?

CodeBuild esegue build in ambienti di compilazione che utilizzano immagini Docker. Queste immagini Docker devono essere archiviate in un tipo di repository supportato da CodeBuild. Questi includono il repository di immagini CodeBuild Docker, Docker Hub e Amazon Elastic Container Registry (Amazon ECR). Per ulteriori informazioni sull'archivio di immagini CodeBuild Docker, consulta [Immagini Docker fornite da CodeBuild](#)

4. Hai bisogno di AWS risorse che non vengono fornite automaticamente da? CodeBuild In caso affermativo, di quali politiche di sicurezza hanno bisogno queste risorse? Ad esempio, potrebbe

essere necessario modificare il ruolo del CodeBuild servizio CodeBuild per consentire l'utilizzo di tali risorse.

5. Vuoi CodeBuild lavorare con il tuo VPC? In tal caso, sono necessari l'ID VPC, la sottorete IDs e il gruppo di sicurezza per la configurazione del IDs VPC. Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Dopo aver risposto a queste domande, disporrai delle impostazioni e delle risorse necessarie per eseguire con successo una compilazione. Per eseguire la compilazione, è possibile:

- Usa la AWS CodeBuild console, AWS CLI o. AWS SDKs Per ulteriori informazioni, consulta [Esegui le build manualmente](#).
- Crea o identifica una pipeline in AWS CodePipeline, quindi aggiungi un'azione di compilazione o test che indichi di CodeBuild testare automaticamente il codice, eseguire la build o entrambi. Per ulteriori informazioni, consulta [Usa CodeBuild con CodePipeline](#).

Riferimento alle specifiche di costruzione per CodeBuild

Questo argomento fornisce importanti informazioni di riferimento sui file di specifiche di compilazione (buildspec). Un buildspec è una raccolta di comandi di compilazione e impostazioni correlate, in formato YAML, che viene da CodeBuild utilizzata per eseguire una build. Puoi includere un buildspec come parte del codice sorgente oppure puoi definire un buildspec quando crei un progetto di build. Per informazioni sul funzionamento di una specifica di build, consulta [Come CodeBuild funziona](#).

Argomenti

- [Nome del file buildspec e posizione di storage](#)
- [Sintassi buildspec](#)
- [Esempio di buildspec](#)
- [Versioni di buildspec](#)
- [Riferimento buildspec per la compilazione in batch](#)

Nome del file buildspec e posizione di storage

Se includi una buildspec come parte del codice sorgente, per impostazione predefinita il file deve essere denominato `buildspec.yml` e deve trovarsi nella root della directory di origine.

È possibile sostituire il nome e la posizione del file buildspec. Ad esempio, puoi:

- Utilizzare un file buildspec diverso per diverse build nello stesso repository, ad esempio `buildspec_debug.yml` e `buildspec_release.yml`.
- Archiviare un file buildspec in una posizione diversa rispetto alla root della directory di origine, ad esempio `config/buildspec.yml` o in un bucket S3. Il bucket S3 deve trovarsi nella stessa regione del progetto di build. AWS Specificare il file buildspec utilizzando il relativo ARN (ad esempio, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

Puoi specificare solo una buildspec per ogni progetto di compilazione, indipendentemente dal nome del file buildspec.

Per sostituire il nome o la posizione predefiniti del file buildspec o entrambi, eseguire una delle seguenti operazioni:

- Esegui il `update-project` comando AWS CLI `create-project` or, impostando il `buildspec` valore del percorso del file `buildspec` alternativo relativo al valore della variabile di ambiente `integrata`. `CODEBUILD_SRC_DIR` È inoltre possibile eseguire l'equivalente con l'`create project` operazione in. AWS SDKs Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione](#) o [Modifica le impostazioni del progetto di compilazione](#).
- Esegui il AWS CLI `start-build` comando, impostando il `buildspecOverride` valore del percorso del file `buildspec` alternativo relativo al valore della variabile di ambiente `incorporata`. `CODEBUILD_SRC_DIR` È inoltre possibile eseguire l'equivalente con l'`start build` operazione in. AWS SDKs Per ulteriori informazioni, consulta [Esegui le build manualmente](#).
- In un AWS CloudFormation modello, imposta la `BuildSpec` proprietà di `Source` in una risorsa di tipo `AWS::CodeBuild::Project` sul percorso del file `buildspec` alternativo relativo al valore della variabile di ambiente `incorporata`. `CODEBUILD_SRC_DIR` Per ulteriori informazioni, consultate la `BuildSpec` proprietà nel codice [sorgente del AWS CodeBuild progetto nella Guida](#) per l'utente. AWS CloudFormation

Sintassi buildspec

I file `buildspec` devono essere espressi in formato [YAML](#).

Se un comando contiene un carattere, o una stringa di caratteri, che non è supportato da YAML, è necessario racchiudere il comando tra virgolette (""). Il seguente comando è racchiuso tra virgolette perché un segno di due punti (:) seguito da uno spazio non è consentito in YAML. La virgoletta nel comando è `escape` (\").

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\\\",,]//g')"
```

Buildspec presenta la seguente sintassi:

```
version: 0.2

run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
```

```
parameter-store:  
  key: "value"  
  key: "value"  
exported-variables:  
  - variable  
  - variable  
secrets-manager:  
  key: secret-id:json-key:version-stage:version-id  
git-credential-helper: no | yes
```

```
proxy:  
  upload-artifacts: no | yes  
  logs: no | yes
```

```
batch:  
  fast-fail: false | true  
  # build-list:  
  # build-matrix:  
  # build-graph:  
  # build-fanout:
```

```
phases:  
  install:  
    run-as: Linux-user-name  
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |  
    RETRY-count-regex  
    runtime-versions:  
      runtime: version  
      runtime: version  
    commands:  
      - command  
      - command  
    finally:  
      - command  
      - command
```

```
pre_build:  
  run-as: Linux-user-name  
  on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |  
  RETRY-count-regex  
  commands:  
    - command  
    - command  
  finally:
```

- *command*
- *command*

build:

run-as: *Linux-user-name*

on-failure: ABORT | CONTINUE | RETRY | RETRY-*count* | RETRY-*regex* |

RETRY-*count-regex*

commands:

- *command*
- *command*

finally:

- *command*
- *command*

post_build:

run-as: *Linux-user-name*

on-failure: ABORT | CONTINUE | RETRY | RETRY-*count* | RETRY-*regex* |

RETRY-*count-regex*

commands:

- *command*
- *command*

finally:

- *command*
- *command*

reports:

report-group-name-or-arn:

files:

- *location*
- *location*

base-directory: *location*

discard-paths: no | yes

file-format: *report-format*

artifacts:**files:**

- *location*
- *location*

name: *artifact-name*

discard-paths: no | yes

base-directory: *location*

exclude-paths: *excluded paths*

enable-symlinks: no | yes

s3-prefix: *prefix*

secondary-artifacts:

```
artifactIdentifier:
  files:
    - location
    - location
  name: secondary-artifact-name
  discard-paths: no | yes
  base-directory: location
artifactIdentifier:
  files:
    - location
    - location
  discard-paths: no | yes
  base-directory: location
cache:
  key: key
  fallback-keys:
    - fallback-key
    - fallback-key
  action: restore | save
  paths:
    - path
    - path
```

Il file buildspec contiene quanto segue:

version

Mappatura obbligatoria. Rappresenta la versione buildspec. Ti consigliamo di utilizzare 0.2.

Note

Anche se la versione 0.1 è ancora supportata, consigliamo di utilizzare la versione 0.2 ogni volta che è possibile. Per ulteriori informazioni, consulta [Versioni di buildspec](#).

run-as

Sequenza opzionale. Disponibile per gli utenti Linux. Specifica un utente Linux che esegue i comandi in questo file buildspec. `run-as` concede all'utente specificato le autorizzazioni di lettura ed esecuzione. Quando specifichi `run-as` nella parte superiore del file buildspec, viene applicato globalmente a tutti i comandi. Se non vuoi specificare un utente per tutti i comandi del file buildspec,

puoi specificarne uno per i comandi di una fase utilizzando `run-as` in uno dei blocchi `phases`. Se `run-as` non è specificato, tutti i comandi vengono eseguiti come utente `root`.

env

Sequenza opzionale. Rappresenta le informazioni su una o più variabili di ambiente personalizzate.

Note

Per proteggere le informazioni riservate, nei log sono nascoste le seguenti informazioni: CodeBuild

- AWS chiave IDs di accesso. Per ulteriori informazioni, consulta [la sezione Gestione delle chiavi di accesso per gli utenti IAM](#) nella Guida per l'AWS Identity and Access Management utente.
- Stringhe specificate utilizzando l'archivio parametri. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.
- Stringhe specificate utilizzando. Gestione dei segreti AWS Per ulteriori informazioni, consulta [Gestione delle chiavi](#).

env/ shell

Sequenza opzionale. Specifica la shell supportata per i sistemi operativi Linux o Windows.

Per i sistemi operativi Linux, i tag shell supportati sono:

- `bash`
- `/bin/sh`

Per i sistemi operativi Windows, i tag shell supportati sono:

- `powershell.exe`
- `cmd.exe`

env/variables

Obbligatorio se è specificato `env` e si desidera definire variabili di ambiente personalizzate in testo normale. Contiene una mappatura di **key/value** scalari, in cui ogni mappatura rappresenta

una singola variabile di ambiente personalizzata in testo semplice. *key* è il nome della variabile di ambiente personalizzata ed *value* è il valore di quella variabile.

Important

Sconsigliamo vivamente la memorizzazione di valori sensibili nelle variabili di ambiente. Le variabili di ambiente possono essere visualizzate in testo semplice utilizzando strumenti come la CodeBuild console e il AWS CLI. Per i valori sensibili, consigliamo di utilizzare la mappatura `parameter-store` o `secrets-manager`, come descritto più avanti in questa sezione.

Tutte le variabili di ambiente impostate sostituiscono quelle esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata `MY_VAR` con un valore di `my_value`, e si imposta una variabile di ambiente denominata `MY_VAR` con un valore di `other_value`, il valore `my_value` viene sostituito da `other_value`. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata `PATH` con un valore di `/usr/local/sbin:/usr/local/bin`, e si imposta una variabile di ambiente denominata `PATH` con un valore di `$PATH:/usr/share/ant/bin`, il valore di `/usr/local/sbin:/usr/local/bin` viene sostituito dal valore letterale `$PATH:/usr/share/ant/bin`.

Non impostare variabili di ambiente con nomi che iniziano con `CODEBUILD_`. Questo prefisso è riservato per l'uso interno.

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità. Al momento della creazione di una compilazione è possibile aggiungere o sostituire variabili di ambiente. Per ulteriori informazioni, consulta [Esegui AWS CodeBuild le build manualmente](#).
- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità. Quando crei o modifichi un progetto, puoi aggiungere variabili di ambiente a livello di progetto. Per ulteriori informazioni, consultare [Crea un progetto di compilazione in AWS CodeBuild](#) e [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).
- Il valore della dichiarazione `buildspec` ha la minima priorità.

env/parameter-store

Obbligatorio se `env` è specificato e desideri recuperare variabili di ambiente personalizzate archiviate in Amazon EC2 Systems Manager Parameter Store. Contiene una mappatura di `key/value` scalars, in cui ogni mappatura rappresenta una singola variabile di ambiente personalizzata memorizzata in Amazon EC2 Systems Manager Parameter Store. `key` è il nome che usi più avanti nei comandi di compilazione per fare riferimento a questa variabile di ambiente personalizzata ed `value` è il nome della variabile di ambiente personalizzata memorizzata in Amazon EC2 Systems Manager Parameter Store. Per memorizzare valori sensibili, consulta [Systems Manager Parameter Store](#) and [Walkthrough: Create and test a String parameter \(console\)](#) nella Amazon EC2 Systems Manager User Guide.

Important

CodeBuild Per consentire il recupero delle variabili di ambiente personalizzate archiviate in Amazon EC2 Systems Manager Parameter Store, devi aggiungere `ssm:GetParameters` l'azione al CodeBuild tuo ruolo di servizio. Per ulteriori informazioni, consulta [Consenti CodeBuild di interagire con altri servizi AWS](#).

Tutte le variabili di ambiente recuperate da Amazon EC2 Systems Manager Parameter Store sostituiscono le variabili di ambiente esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata `MY_VAR` con un valore di `my_value` e si recupera una variabile di ambiente denominata `MY_VAR` con un valore di `other_value`, il valore `my_value` viene sostituito da `other_value`. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata `PATH` con un valore di `/usr/local/sbin:/usr/local/bin` e si recupera una variabile di ambiente denominata `PATH` con un valore di `$PATH:/usr/share/ant/bin`, il valore di `/usr/local/sbin:/usr/local/bin` viene sostituito dal valore letterale `$PATH:/usr/share/ant/bin`.

Non archiviare variabili di ambiente con nomi che iniziano con `CODEBUILD_`. Questo prefisso è riservato per l'uso interno .

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità. Al momento della creazione di una compilazione è possibile aggiungere o sostituire variabili di ambiente. Per ulteriori informazioni, consulta [Esegui AWS CodeBuild le build manualmente](#).

- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità. Quando crei o modifichi un progetto, puoi aggiungere variabili di ambiente a livello di progetto. Per ulteriori informazioni, consultare [Crea un progetto di compilazione in AWS CodeBuild](#) e [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).
- Il valore della dichiarazione buildspec ha la minima priorità.

env/secrets-manager

Obbligatorio se desideri recuperare le variabili di ambiente personalizzate archiviate in. Gestione dei segreti AWS Specificate un Secrets Manager reference-key utilizzando il seguente schema:

<key>: <secret-id>:<json-key>:<version-stage>:<version-id>

<key>

(Obbligatorio) Il nome della variabile di ambiente locale. Usa questo nome per accedere alla variabile durante la compilazione.

<secret-id>

(Obbligatorio) Il nome o Amazon Resource Name (ARN) che funge da identificatore univoco per il segreto. Per accedere a un segreto nel tuo account AWS , devi semplicemente specificare il nome segreto. Per accedere a un segreto in un altro AWS account, specifica l'ARN segreto.

<json-key>

(Facoltativo) Specificate il nome della chiave della coppia chiave-valore di Secrets Manager di cui desiderate recuperare il valore. Se non si specifica a json-key, CodeBuild recupera l'intero testo segreto.

<version-stage>

(Facoltativo) Specificate la versione segreta che desiderate recuperare tramite l'etichetta temporanea allegata alla versione. Le etichette temporanee sono utilizzate per tenere traccia di differenti versioni durante il processo di rotazione. Se utilizzi version-stage, non specificare version-id. Se non specifichi una fase o un ID di versione, l'impostazione predefinita è recuperare la versione con il valore di fase di versione AWSCURRENT.

<version-id>

(Facoltativo) Specificate l'identificatore univoco della versione del segreto che desiderate utilizzare. Se specifichi `version-id`, non specificare `version-stage`. Se non specifichi una fase o un ID di versione, l'impostazione predefinita è recuperare la versione con il valore di fase di versione `AWSCURRENT`.

Nell'esempio seguente, `TestSecret` è il nome della coppia chiave-valore memorizzata in Secrets Manager. La chiave per `TestSecret` è `MY_SECRET_VAR`. Si accede alla variabile durante la compilazione utilizzando il `LOCAL_SECRET_VAR` nome.

```
env:
  secrets-manager:
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

env/exported-variables

Mappatura opzionale. Utilizzato per elencare le variabili di ambiente che si desidera esportare. Specificare il nome di ogni variabile che si desidera esportare su una riga separata sotto `exported-variables`. La variabile che si desidera esportare deve essere disponibile nel contenitore durante la compilazione. La variabile esportata può essere una variabile di ambiente.

Le variabili di ambiente esportate vengono utilizzate insieme AWS CodePipeline a per esportare le variabili di ambiente dalla fase di compilazione corrente alle fasi successive della pipeline. Per ulteriori informazioni, consulta [Lavorare con le variabili nella Guida](#) per l'AWS CodePipeline utente.

Durante una compilazione, il valore di una variabile è disponibile a partire dalla fase `install`. Può essere aggiornato tra l'inizio della fase `install` e la fine della fase `post_build`. Al termine della fase `post_build`, il valore delle variabili esportate non può cambiare.

Note

Non è possibile esportare quanto segue:

- I segreti di Amazon EC2 Systems Manager Parameter Store specificati nel progetto di compilazione.
- Secrets Manager: segreti specificati nel progetto di compilazione

- Variabili di ambiente che iniziano con `AWS_`.

env/ git-credential-helper

Mappatura opzionale. Usato per indicare se CodeBuild utilizza il suo helper di credenziali Git per fornire le credenziali Git. `yes` se è usato. In caso contrario, `no` o non specificato. Per ulteriori informazioni, consulta [gitcredentials](#) sul sito Web Git.

Note

`git-credential-helper` non è supportato per le compilazioni attivate da un webhook per un repository Git pubblico.

proxy

Sequenza opzionale. Utilizzato per rappresentare le impostazioni se esegui la compilazione in un server proxy esplicito. Per ulteriori informazioni, consulta [Esegui CodeBuild in un server proxy esplicito](#).

proxy/upload-artifacts

Mappatura opzionale. Imposta su `yes` se desideri che la compilazione in un server proxy esplicito carichi artefatti. Il valore predefinito è `no`.

proxy/logs

Mappatura opzionale. Imposta su `yes` per la creazione di un server proxy esplicito per la creazione di CloudWatch registri. Il valore predefinito è `no`.

phases

Sequenza obbligatoria. Rappresenta i comandi CodeBuild eseguiti durante ogni fase della compilazione.

Note

Nella versione 0.1 di `buildspec`, CodeBuild esegue ogni comando in un'istanza separata della shell predefinita nell'ambiente di compilazione. Questo significa che ogni comando viene

eseguito separatamente da tutti gli altri comandi. Pertanto, come impostazione predefinita, non è possibile eseguire un singolo comando che si basa sullo stato di qualsiasi comando precedente (per esempio, modificare le directory o impostare le variabili di ambiente). Per aggirare questo limite, suggeriamo di utilizzare la versione 0.2, che risolve questo problema. Se devi utilizzare la versione 0.1 di buildspec, ti consigliamo gli approcci indicati in [Shell e comandi negli ambienti di compilazione](#).

phases/*/run-as

Sequenza opzionale. Viene utilizzato in una fase di compilazione per specificare un utente Linux che esegue i comandi. Se `run-as` è specificato anche a livello globale per tutti i comandi nella parte superiore del file buildspec, l'utente a livello di fase ha la priorità. Ad esempio, se globalmente `run-as` specifica User-1 e per la fase `install` solo una istruzione `run-as` specifica User-2, tutti i comandi nel file buildspec vengono eseguiti come User-1 eccetto i comandi nella fase `install` che vengono eseguiti come User-2.

phases/*/ in caso di errore

Sequenza opzionale. Specifica l'azione da intraprendere se si verifica un errore durante la fase. Può essere uno dei seguenti valori:

- **ABORT**- Interrompi la compilazione.
- **CONTINUE**- Passa alla fase successiva.
- **RETRY**- Riprova la compilazione fino a 3 volte con un messaggio di errore che corrisponde all'espressione `. * regolare`.
- **RETRY-*count***- Riprova la compilazione per un numero specificato di volte, come rappresentato da un messaggio *count* di errore che corrisponde all'espressione regolare. `. *` Nota che *count* deve essere compreso tra 0 e 100. Ad esempio, i valori validi includono `RETRY-4` e `RETRY-8`.
- **RETRY-*regex***- Riprova la compilazione fino a 3 volte e usala *regex* per includere un'espressione regolare che corrisponda a un messaggio di errore specificato. Ad esempio, i valori validi includono `Retry-. *Error: Unable to connect to database.*` e `RETRY-invalid+`.
- **RETRY-*count-regex***- Riprova la build per un numero di volte specificato, come rappresentato da *count*. Nota che *count* deve essere compreso tra 0 e 100. Puoi anche usare *regex* per includere un'espressione regolare che corrisponda al messaggio di errore. Ad esempio, i valori validi includono `Retry-3-. *connection timed out.*` e `RETRY-8-invalid+`.

Se questa proprietà non è specificata, il processo di fallimento segue le fasi di transizione illustrate in [Transizioni delle fasi della compilazione](#).

 Important

L'on-failureattributo non è supportato quando si utilizza l'elaborazione Lambda o la capacità riservata. Questo attributo funziona solo con le immagini di calcolo EC2 fornite da CodeBuild

phases/*/ finalmente

Blocco opzionale. I comandi specificati in un `finally` blocco vengono eseguiti dopo i comandi nel blocco. `commands` I comandi in un `finally` blocco vengono eseguiti anche se un comando nel `commands` blocco ha esito negativo. Ad esempio, se il `commands` blocco contiene tre comandi e il primo ha esito negativo, CodeBuild ignora i due comandi rimanenti ed esegue tutti i comandi del `finally` blocco. La fase va a buon fine quando l'esecuzione di tutti i comandi nei blocchi `commands` e `finally` vengono completata. Se uno qualunque dei comandi nella fase non va a buon fine, la fase non va a buon fine:

I nomi delle fasi di build ammessi sono:

phases/install

Sequenza opzionale. Rappresenta gli eventuali comandi che vengono CodeBuild eseguiti durante l'installazione. Ti consigliamo di utilizzare la fase `install` solo per installare pacchetti nell'ambiente di build. Ad esempio, è possibile utilizzare questa fase per installare un framework di test del codice come Mocha o RSpec.

phases/install/runtime-versions

Sequenza opzionale. Una versione runtime è supportata con l'immagine standard di Ubuntu 5.0 o successiva e l'immagine standard Amazon Linux 2 4.0 o successiva. Se specificata, almeno un runtime deve essere incluso in questa sezione. Specificate un runtime utilizzando una versione specifica, una versione principale seguita da `.x` da specificare che CodeBuild utilizza quella versione principale con la versione secondaria più recente o `latest` per utilizzare la versione principale e secondaria più recente (ad esempio `ruby: 3.2`, `nodejs: 18.x`, `java: latest`). Puoi specificare il runtime utilizzando un numero o una variabile di ambiente. Ad esempio, se utilizzi l'immagine standard di Amazon Linux 2 4.0, quanto

segue specifica che è installata la versione 17 di Java, l'ultima versione secondaria di python versione 3 e una versione contenuta in una variabile di ambiente di Ruby. Per ulteriori informazioni, consulta [Immagini Docker fornite da CodeBuild](#).

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

È possibile specificare uno o più runtime nella sezione `runtime-versions` del file `buildspec`. Se il runtime dipende da un altro runtime, è anche possibile specificarne il runtime dipendente nel file `buildspec`. Se non specifichi alcun runtime nel file `buildspec`, CodeBuild sceglie i runtime predefiniti disponibili nell'immagine che usi. Se specificate uno o più runtime, utilizza solo tali runtime. CodeBuild Se non viene specificato un runtime dipendente, CodeBuild tenta di scegliere automaticamente il runtime dipendente.

Se due runtime specificati sono in conflitto, la compilazione ha esito negativo. Ad esempio, `android: 29` e `java: openjdk11` sono in conflitto, perciò, se vengono specificati entrambi, la compilazione non riesce.

Per ulteriori informazioni sui runtime disponibili, vedere [Runtime disponibili](#).

Note

Se specifichi una `runtime-versions` sezione e usi un'immagine diversa da Ubuntu Standard Image 2.0 o versione successiva o dall'immagine standard Amazon Linux 2 (AL2) 1.0 o successiva, la build emette l'avviso "Skipping install of runtimes. Runtime version selection is not supported by this build image.»

phases/install/commands

Sequenza opzionale. Contiene una sequenza di scalari, in cui ogni scalare rappresenta un singolo comando che CodeBuild viene eseguito durante l'installazione. CodeBuild esegue ogni comando, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

phases/pre_build

Sequenza opzionale. Rappresenta gli eventuali comandi CodeBuild eseguiti prima della compilazione. Ad esempio, potresti utilizzare questa fase per accedere ad Amazon ECR o installare npm dependencies.

phases/pre_build/commands

Sequenza obbligatoria se viene specificato `pre_build`. Contiene una sequenza di scalari, in cui ogni scalare rappresenta un singolo comando che CodeBuild viene eseguito prima della compilazione. CodeBuild esegue ogni comando, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

phases/build

Sequenza opzionale. Rappresenta gli eventuali comandi CodeBuild eseguiti durante la compilazione. Ad esempio, è possibile utilizzare questa fase per eseguire Mocha o RSpec sbt.

phases/build/commands

Obbligatorio se `build` specificato. Contiene una sequenza di scalari, in cui ogni scalare rappresenta un singolo comando che CodeBuild viene eseguito durante la compilazione.

CodeBuild esegue ogni comando, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

phases/post_build

Sequenza opzionale. Rappresenta gli eventuali comandi che vengono CodeBuild eseguiti dopo la compilazione. Ad esempio, puoi usare Maven per impacchettare gli artefatti di compilazione in un file JAR o WAR oppure puoi inserire un'immagine Docker in Amazon ECR. Quindi puoi inviare una notifica di build tramite Amazon SNS.

phases/post_build/commands

Obbligatorio `post_build` se specificato. Contiene una sequenza di scalari, in cui ogni scalare rappresenta un singolo comando che CodeBuild viene eseguito dopo la compilazione.

CodeBuild esegue ogni comando, uno alla volta, nell'ordine elencato, dall'inizio alla fine.

reports

report-group-name-or-arn

Sequenza opzionale. Specifica il gruppo di report a cui vengono inviati i report. Un progetto può annoverare al massimo cinque gruppi di report. Specificare l'ARN di un gruppo di report

esistente o il nome di un nuovo gruppo di report. Se specificate un nome, CodeBuild crea un gruppo di report utilizzando il nome del progetto e il nome specificato nel formato. `<project-name>-<report-group-name>` Il nome del gruppo di report può anche essere impostato utilizzando una variabile di ambiente nel buildspec, ad esempio. `$REPORT_GROUP_NAME` Per ulteriori informazioni, consulta [Denominazione dei gruppi di report](#).

`reports/<report-group>/files`

Sequenza obbligatoria. Rappresenta le collocazioni dei dati grezzi afferenti ai risultati di test generati dal report. Contiene una sequenza di scalari, con ogni scalare che rappresenta una posizione separata in cui è CodeBuild possibile trovare i file di test, rispetto alla posizione di build originale o, se impostata, a. `base-directory` Le posizioni possono includere quanto segue:

- Un solo file (ad esempio `my-test-report-file.json`).
- Un solo file in una sottodirectory (ad esempio `my-subdirectory/my-test-report-file.json` o `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/*'` rappresenta tutti i file in modo ricorsivo.
- `my-subdirectory/*` rappresenta tutti i file in una sottodirectory denominata. `my-subdirectory`
- `my-subdirectory/**/*` rappresenta tutti i file in modo ricorsivo a partire da una sottodirectory denominata. `my-subdirectory`

`reports/<report-group>/file-format`

Mappatura opzionale. Rappresenta il formato del file di report. In assenza di specifica, viene utilizzato JUNITXML. Questo valore non fa distinzione tra maiuscole e minuscole. I valori possibili sono:

Rapporti sui test

CUCUMBERJSON

Cucumber JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

 JUnit XML 3

TESTNGXML

 TestNG XML

VISUALSTUDIOTRX

 Visual Studio TRX

Report di copertura del codice

CLOVERXML

 Clover XML

COBERTURAXML

 Cobertura XML

JACOCOXML

 JaCoCo XML

SIMPLECOV

 SimpleCov JSON

 Note

CodeBuild [accetta i report sulla copertura del codice JSON generati da simplecov, non da simplecov-json.](#)

reports/<report-group>/base-directory

Mappatura opzionale. Rappresenta una o più directory di primo livello, relative alla posizione di build originale, che vengono CodeBuild utilizzate per determinare dove trovare i file di test non elaborati.

reports/<report-group>/discard-paths

Opzionale. Specifica se le directory dei file di report vengono appiattite nell'output. Se non è specificato o contiene no, i file di report vengono generati con la loro struttura di directory intatta. Se contiene yes, tutti i file di test vengono inseriti nella stessa directory di output. Ad esempio, se

un percorso di un risultato del test è `com/myapp/mytests/TestResult.xml`, specificando `yes` questo file verrà posizionato in `/TestResult.xml`.

artefatti

Sequenza opzionale. Rappresenta informazioni su dove CodeBuild è possibile trovare l'output della build e su come CodeBuild prepararlo per il caricamento nel bucket di output S3. Questa sequenza non è necessaria se, ad esempio, stai creando e inviando un'immagine Docker ad Amazon ECR oppure esegui test unitari sul tuo codice sorgente, ma non lo compili.

Note

I metadati di Amazon S3 hanno un' intestazione denominata `x-amz-meta-codebuild-buildarn` che contiene la CodeBuild build che pubblica gli `buildArn` artefatti su Amazon S3. `buildArn`Viene aggiunto per consentire il tracciamento della fonte per le notifiche e per fare riferimento alla build da cui viene generato l'artefatto.

artifacts/files

Sequenza obbligatoria. Rappresenta le posizioni contenenti gli artefatti di output della build nell'ambiente della build. Contiene una sequenza di scalari, ognuno dei quali rappresenta una posizione diversa dove CodeBuild può trovare gli artefatti dell'output della build rispetto alla posizione della build originale o, se impostata, alla directory di base. Le posizioni possono includere quanto segue:

- Un solo file (ad esempio `my-file.jar`).
- Un solo file in una sottodirectory (ad esempio `my-subdirectory/my-file.jar` o `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` rappresenta tutti i file in modo ricorsivo.
- `my-subdirectory/*` rappresenta tutti i file in una sottodirectory denominata. `my-subdirectory`
- `my-subdirectory/**/*` rappresenta tutti i file in modo ricorsivo a partire da una sottodirectory denominata. `my-subdirectory`

Quando specificate le posizioni degli artefatti di output della build, CodeBuild potete individuare la posizione di compilazione originale nell'ambiente di compilazione. Non occorre anteporre le

posizioni dell'output dell'artefatto della build al percorso verso la posizione della build originale, né specificare `./` o valori simili. Se vuoi conoscere il percorso verso la posizione, puoi eseguire un comando come `echo $CODEBUILD_SRC_DIR` durante una build. La posizione di ciascun ambiente di build può essere leggermente diversa.

artifacts/name

Nome opzionale. Specifica un nome per l'artefatto della build. Questo nome si usa quando una delle seguenti affermazioni è vera.

- Utilizzate l' CodeBuild API per creare le vostre build e il `overrideArtifactName` flag viene impostato sull'`ProjectArtifacts` oggetto quando un progetto viene aggiornato, creato un progetto o viene avviata una build.
- Si utilizza la CodeBuild console per creare le build, viene specificato un nome nel file `buildspec` e si seleziona `Abilita il controllo delle versioni semantiche` quando si crea o si aggiorna un progetto. Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

Puoi specificare un nome nel file `buildspec` calcolato in fase di creazione. Il nome specificato nel file `buildspec` utilizza il linguaggio di comando Shell. Ad esempio, è possibile aggiungere una data e un'ora al nome dell'artefatto in modo che sia sempre univoco. I nomi di artefatto univoci impediscono che gli artefatti vengano sovrascritti. Per ulteriori informazioni, consulta [Linguaggio di comando Shell](#).

- Questo è un esempio di un nome di artefatto a cui viene aggiunta la data di creazione dell'artefatto.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

- Questo è un esempio di nome di artefatto che utilizza una variabile di ambiente. CodeBuild Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).

```
version: 0.2
phases:
  build:
```

```

    commands:
      - rspec HelloWorld_spec.rb
  artifacts:
    files:
      - '**/*'
    name: myname-$AWS_REGION

```

- Questo è un esempio di nome di artefatto che utilizza una variabile di CodeBuild ambiente a cui è aggiunta la data di creazione dell'artefatto.

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
  artifacts:
    files:
      - '**/*'
    name: $AWS_REGION-$(date +%Y-%m-%d)

```

È possibile aggiungere informazioni sul percorso al nome in modo che gli artefatti denominati vengano inseriti nelle directory in base al percorso contenuto nel nome. In questo esempio, gli artefatti della build vengono inseriti nell'output sotto. `builds/<build number>/my-artifacts`

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
  artifacts:
    files:
      - '**/*'
    name: builds/$CODEBUILD_BUILD_NUMBER/my-artifacts

```

artifacts/discard-paths

Opzionale. Specifica se le directory degli artefatti di build vengono appiattite nell'output. Se non è specificato, o contiene `no`, gli artefatti di build vengono generati con la loro struttura di directory intatta. Se contiene `yes`, tutti gli artefatti di build vengono inseriti nella stessa directory di output.

Ad esempio, se un percorso di un file nell'artefatto di output della build è `com/mycompany/app/HelloWorld.java`, specificando `yes` il file verrà posizionato in `/HelloWorld.java`.

`artifacts/base-directory`

Mappatura opzionale. Rappresenta una o più directory di primo livello, relative alla posizione di compilazione originale, che CodeBuild vengono utilizzate per determinare quali file e sottodirectory includere nell'elemento di output della build. I valori validi includono:

- Una sola directory di primo livello (ad esempio `my-directory`).
- `'my-directory*'` rappresenta tutte le directory di primo livello i cui nomi iniziano con `my-directory`.

Le directory di primo livello corrispondenti non sono incluse nell'artefatto dell'output della build, ma solo i file e le sottodirectory.

Puoi utilizzare `files` e `discard-paths` per limitare ulteriormente i file e le sottodirectory inclusi. Ad esempio, per la seguente struttura di directory:

```
.  
### my-build-1  
#   ### my-file-1.txt  
### my-build-2  
    ### my-file-2.txt  
    ### my-subdirectory  
        ### my-file-3.txt
```

E per la seguente sequenza `artifacts`:

```
artifacts:  
  files:  
    - '*/my-file-3.txt'  
  base-directory: my-build-2
```

Nell'artefatto di output della build sarebbero compresi la seguente sottodirectory e il seguente file:

```
.  
### my-subdirectory  
    ### my-file-3.txt
```

Mentre per la seguente sequenza `artifacts`:

```
artifacts:
  files:
    - '**/*'
  base-directory: 'my-build*'
  discard-paths: yes
```

Nell'artefatto di output della build sarebbero compresi i seguenti file:

```
.
### my-file-1.txt
### my-file-2.txt
### my-file-3.txt
```

artefatti/ percorsi di esclusione

Mappatura opzionale. Rappresenta uno o più percorsi, relativi a, che CodeBuild escluderanno gli `base-directory` artefatti della build. L'asterisco (*) individua zero o più caratteri di un componente del nome entro i limiti della cartella. Un doppio asterisco (**) corrisponde a zero o più caratteri di un componente del nome in tutte le directory.

Di seguito sono riportati alcuni esempi di percorsi di esclusione:

- Per escludere un file da tutte le directory: `"**/file-name/**/*"`
- Per escludere tutte le cartelle dot: `"**/.*/**/*"`
- Per escludere tutti i file dot: `"**/*. *"`

artefatti/ enable-symlinks

Opzionale. Se il tipo di output è ZIP, specifica se i collegamenti simbolici interni vengono conservati nel file ZIP. Se `contieneyes`, tutti i collegamenti simbolici interni nel codice sorgente verranno conservati nel file ZIP degli artefatti.

artefatti/ prefisso s3

Opzionale. Specifica un prefisso utilizzato quando gli artefatti vengono emessi in un bucket Amazon S3 e il tipo di namespace è. `BUILD_ID` Quando viene utilizzato, il percorso di output nel bucket è. `<s3-prefix>/<build-id>/<name>.zip`

artifacts/secondary-artifacts

Sequenza opzionale. Rappresenta una o più definizioni di artefatto come mappatura tra un identificatore dell'artefatto e una sua definizione. Ciascun identificatore di artefatto in questo

blocco deve corrispondere a un artefatto definito nell'attributo `secondaryArtifacts` del progetto. Ogni diversa definizione presenta la stessa sintassi del blocco `artifacts` illustrato in precedenza.

 Note

La [artifacts/files](#) sequenza è sempre obbligatoria, anche quando sono definiti solo artefatti secondari.

Ad esempio, se un progetto presenta la seguente struttura:

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

Il `buildspec` avrà il seguente aspetto:

```
version: 0.2

phases:
  build:
    commands:
      - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
```



```
artifact1:
  files:
    - directory/file1
  name: secondary-artifact-name-1
artifact2:
  files:
    - directory/file2
  name: secondary-artifact-name-2
```

cache

Sequenza opzionale. Rappresenta informazioni su dove CodeBuild è possibile preparare i file per il caricamento della cache in un bucket di cache S3. Questa sequenza non è obbligatoria se il tipo di cache del progetto è No Cache.

cache/ chiave

Sequenza opzionale. Rappresenta la chiave primaria utilizzata per la ricerca o il ripristino di una cache. CodeBuild corrisponde esattamente alla chiave primaria.

Ecco un esempio della chiave:

```
key: npm-key-${codebuild-hash-files package-lock.json} }
```

cache/ chiavi di riserva

Sequenza opzionale. Rappresenta un elenco di chiavi di fallback utilizzate in sequenza quando non è possibile trovare una cache utilizzando la chiave primaria. Sono supportate fino a cinque chiavi di fallback, ognuna delle quali viene abbinata mediante una ricerca per prefisso. Questa sequenza verrà ignorata se la chiave non viene fornita.

Ecco un esempio per le chiavi di riserva:

```
fallback-keys:
  - npm-key-${codebuild-hash-files package-lock.json} }
  - npm-key-
  - npm-
```

cache/ azione

Sequenza opzionale. Specifica l'azione da eseguire sulla cache. I valori validi includono:

- `restore` che ripristina solo la cache senza salvare gli aggiornamenti.
- `save` che salva solo la cache senza ripristinare una versione precedente.

Se non viene fornito alcun valore, l' CodeBuild impostazione predefinita esegue sia il ripristino che il salvataggio.

cache/paths

Sequenza obbligatoria. Rappresenta le posizioni della cache. Contiene una sequenza di scalari, ciascuno dei quali rappresenta una posizione separata in cui è CodeBuild possibile trovare gli artefatti di output della build, rispetto alla posizione di build originale o, se impostata, alla directory di base. Le posizioni possono includere quanto segue:

- Un solo file (ad esempio `my-file.jar`).
- Un solo file in una sottodirectory (ad esempio `my-subdirectory/my-file.jar` o `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` rappresenta tutti i file in modo ricorsivo.
- `my-subdirectory/*` rappresenta tutti i file in una sottodirectory denominata. `my-subdirectory`
- `my-subdirectory/**/*` rappresenta tutti i file in modo ricorsivo a partire da una sottodirectory denominata. `my-subdirectory`

Important

Poiché una dichiarazione del file `buildspec` deve essere uno YAML valido, in una dichiarazione `buildspec` è importante la spaziatura. Se il numero di spazi nella dichiarazione `buildspec` non è valido, la build potrebbe non andare a buon fine immediatamente. È possibile usare un convalidatore YAML per verificare se le dichiarazioni `buildspec` sono YAML valide. Se si utilizza il AWS CLI, or the AWS SDKs per dichiarare un `buildspec` quando si crea o si aggiorna un progetto di compilazione, il `buildspec` deve essere una singola stringa espressa in formato YAML, insieme agli spazi bianchi e ai caratteri di escape di nuova riga richiesti. Un esempio è riportato nella sezione successiva.

Se utilizzate le AWS CodePipeline console CodeBuild or invece di un file `buildspec.yml`, potete inserire comandi solo per la fase. `build` Invece di utilizzare la sintassi precedente, si elencano in una sola riga tutti i comandi da eseguire in fase di build. In caso di più comandi, separa ogni comando con `&&` (ad esempio `mvn test && mvn package`).

Puoi usare CodeBuild or CodePipeline consoles invece di un file `buildspec.yml` per specificare le posizioni degli artefatti di output della build nell'ambiente di compilazione.

Invece di utilizzare la sintassi precedente, si elencano in una sola riga tutte le posizioni. Per più posizioni, separarne ognuna con una virgola (per esempio, `buildspec.yml`, `target/my-app.jar`).

Esempio di buildspec

Ecco un esempio di file `buildspec.yml`.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
```

```
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`
```

```

reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
    files:
      - "**/*"
    base-directory: 'target/tests/reports'
    discard-paths: no
  reportGroupCucumberJson:
    files:
      - 'cucumber/target/cucumber-tests.xml'
    discard-paths: yes
    file-format: CUCUMBERJSON # default is JUNITXML
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
  secondary-artifacts:
    artifact1:
      files:
        - target/artifact-1.0.jar
      discard-paths: yes
    artifact2:
      files:
        - target/artifact-2.0.jar
      discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'

```

Ecco un esempio del precedente buildspec, espresso come una singola stringa, da utilizzare con o. AWS CLI AWS SDKs

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-amd64\n\nparameter-store:\n  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n\nphases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n      - apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This always runs even if the update or install command fails\n\n  pre_build:\n    commands:\n      - echo Entered the pre_build phase...\n      - docker login -u User -p $LOGIN_PASSWORD\n    finally:\n      - echo This always runs even if the login command fails\n\n  build:\n    commands:\n      - echo Entered the build phase...\n      - echo Build started on `date`\n      - mvn install\n    finally:\n      - echo This always runs even if the install command fails\n\n  post_build:\n    commands:\n      - echo Entered the post_build phase...\n      - echo Build completed on `date`\n\nreports:

```

```
\n  reportGroupJUnitXml:\n    files:\n      - \"**/*\" \n    base-directory: 'target/
tests/reports'\n    discard-paths: false\n  reportGroupCucumberJson:\n    files:\n
  - 'cucumber/target/cucumber-tests.xml'\n    file-format: CUCUMBERJSON\n\nartifacts:\n
files:\n  - target/messageUtil-1.0.jar\n  discard-paths: yes\n  secondary-artifacts:
\n  artifact1:\n    files:\n      - target/messageUtil-1.0.jar\n    discard-
paths: yes\n  artifact2:\n    files:\n      - target/messageUtil-1.0.jar\n
discard-paths: yes\n  cache:\n    paths:\n      - '/root/.m2/**/*'
```

Ecco un esempio dei comandi della build fase, da utilizzare con le console or. CodeBuild CodePipeline

```
echo Build started on `date` && mvn install
```

In questi esempi:

- È impostata una variabile di ambiente personalizzata, in testo normale, con chiave JAVA_HOME e valore /usr/lib/jvm/java-8-openjdk-amd64.
- Una variabile di ambiente personalizzata denominata dockerLoginPassword you memorizzata in Amazon EC2 Systems Manager Parameter Store viene referenziata più avanti nei comandi di compilazione utilizzando la chiave. LOGIN_PASSWORD
- Non è possibile modificare questi nomi di fase della build. I comandi eseguiti in questo esempio sono apt-get update -y and apt-get install -y maven (per installare Apache Maven), mvn install (per compilare, testare e impacchettare il codice sorgente in un elemento di output di build e installare l'artefatto di output della build nel suo repository interno), docker login (per accedere a Docker con la password che corrisponde al valore della variabile di ambiente personalizzata impostata in Amazon dockerLoginPassword EC2 Systems Manager Parameter Store) e diversi comandi. echo I echo comandi sono inclusi qui per mostrare come vengono CodeBuild eseguiti i comandi e l'ordine in cui vengono eseguiti.
- files rappresenta i file da caricare nella posizione dell'output della build. In questo esempio, CodeBuild carica il singolo filemessageUtil-1.0.jar. Il file messageUtil-1.0.jar si trova nella relativa directory denominata target nell'ambiente di build. Poiché è specificato discard-paths: yes, messageUtil-1.0.jar viene caricato direttamente (non su una directory target intermedia). Il nome del file messageUtil-1.0.jar e il nome della relativa directory target sono basati sul modo in cui Apache Maven crea e archivia artefatti di output della build unicamente per questo esempio. Nei tuoi scenari, questi nomi di file e directory saranno differenti.
- reports rappresenta due gruppi di report che generano report durante la compilazione:

- `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` specifica l'ARN di un gruppo di report. I risultati dei test generati dal framework di test si trovano nella directory `target/tests/reports`. Il formato dei file è `JUnitXml` e il percorso non viene rimosso dai file contenenti i risultati dei test.
- `reportGroupCucumberJson` specifica un nuovo gruppo di report. Se il nome del progetto è `my-project`, nel momento in cui si esegue una compilazione viene creato un gruppo di report denominato `my-project-reportGroupCucumberJson`. I risultati dei test generati dal framework di test si trovano in `cucumber/target/cucumber-tests.xml`. Il formato dei file di test è `CucumberJson` e il percorso non viene rimosso dai file contenenti i risultati dei test.

Versioni di buildspec

La seguente tabella elenca le versioni delle buildspec e le modifiche tra le versioni.

Versione	Modifiche
0.2	• <code>environment_variables</code> è stato rinominato in <code>env</code>. • <code>plaintext</code> è stato rinominato in <code>variables</code>. • La proprietà <code>type</code> per <code>artifacts</code> è obsoleta. • Nella versione 0.1, AWS CodeBuild esegue ogni comando di compilazione in un'istanza a separata della shell predefinita nell'ambiente di compilazione. Nella versione 0.2, CodeBuild esegue tutti i comandi di compilazione nella stessa istanza della shell predefinita nell'ambiente di compilazione.
0.1	Questa è la definizione iniziale del formato della specifica di build.

Riferimento buildspec per la compilazione in batch

Questo argomento contiene il riferimento buildspec per le proprietà di compilazione in batch.

batch

Mappatura opzionale. Le impostazioni di compilazione in batch per il progetto.

batch/ fast-fail

Facoltativo. Specifica il comportamento della compilazione in batch quando una o più attività di compilazione falliscono.

false

Il valore predefinito. Tutte le build in esecuzione verranno completate.

true

Tutte le build in esecuzione verranno interrotte quando una delle attività di compilazione fallisce.

Per impostazione predefinita, tutte le attività di compilazione in batch vengono eseguite con le impostazioni di compilazione come env ephases, specificate nel file buildspec. È possibile sovrascrivere le impostazioni di build predefinite specificando env valori diversi o un file buildspec diverso nel parametro. batch/<batch-type>/buildspec

Il contenuto della batch proprietà varia in base al tipo di build batch specificato. I possibili tipi di build in batch sono:

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)
- [batch/build-fanout](#)

batch/build-graph

Definisce un grafico di costruzione. Un grafico di compilazione definisce un insieme di attività che dipendono da altre attività del batch. Per ulteriori informazioni, consulta [Costruisci un grafico](#).

Questo elemento contiene una serie di attività di compilazione. Ogni attività di compilazione contiene le seguenti proprietà.

identificatore

Obbligatorio. L'identificatore dell'attività.

buildspec

Facoltativo. Il percorso e il nome del file buildspec da utilizzare per questa attività. Se questo parametro non è specificato, viene utilizzato il file buildspec corrente.

sessione di debug

Facoltativo. Un valore booleano che indica se il debug della sessione è abilitato per questa build in batch. Per ulteriori informazioni sul debug delle sessioni, vedere [Compilazioni di debug con Session Manager](#)

false

Il debug della sessione è disabilitato.

true

Il debug della sessione è abilitato.

dipende da

Facoltativo. Una serie di identificatori di attività da cui dipende questa attività. Questa attività non verrà eseguita fino al completamento di tali attività.

env

Facoltativo. L'ambiente di compilazione sostituisce l'attività. Questo può contenere le seguenti proprietà:

tipo di calcolo

L'identificatore del tipo di calcolo da utilizzare per l'attività. Vedere ComputeType [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) in per i valori possibili.

flotta

L'identificatore della flotta da utilizzare per l'attività. Per ulteriori informazioni, consulta [the section called “Run si basa su flotte a capacità riservata”](#).

image

L'identificatore dell'immagine da utilizzare per l'attività. Vedi Identificatore di immagine in [the section called “Immagini Docker fornite da CodeBuild”](#) per i valori possibili.

modalità privilegiata

Un valore booleano che indica se eseguire il demone Docker all'interno di un contenitore Docker. Imposta su `true` solo se il progetto di compilazione viene utilizzato per creare immagini Docker. In caso contrario, una build che tenta di interagire con il daemon Docker fallisce. L'impostazione predefinita è `false`.

tipo

L'identificatore del tipo di ambiente da utilizzare per l'attività. Vedi Environment type in [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) per i valori possibili.

variabili

Le variabili di ambiente che saranno presenti nell'ambiente di compilazione. Per ulteriori informazioni, consulta [env/variables](#).

Note

Tieni presente che il tipo di calcolo e il parco macchine non possono essere forniti nello stesso identificatore di una singola build.

ignore-failure

Facoltativo. Un valore booleano che indica se un errore di questa operazione di compilazione può essere ignorato.

false

Il valore predefinito. Se questa operazione di compilazione fallisce, la compilazione in batch avrà esito negativo.

true

Se questa operazione di compilazione fallisce, la compilazione in batch può comunque avere successo.

Di seguito è riportato un esempio di una voce buildspec del grafico build:

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

batch/build-list

Definisce un elenco di build. Un elenco di compilazione viene utilizzato per definire una serie di attività eseguite in parallelo. Per ulteriori informazioni, consulta [Crea un elenco](#).

Questo elemento contiene una serie di attività di compilazione. Ogni attività di compilazione contiene le seguenti proprietà.

identificatore

Obbligatorio. L'identificatore dell'attività.

buildspec

Facoltativo. Il percorso e il nome del file buildspec da utilizzare per questa attività. Se questo parametro non è specificato, viene utilizzato il file buildspec corrente.

sessione di debug

Facoltativo. Un valore booleano che indica se il debug della sessione è abilitato per questa build in batch. Per ulteriori informazioni sul debug delle sessioni, vedere. [Compilazioni di debug con Session Manager](#)

false

Il debug della sessione è disabilitato.

true

Il debug della sessione è abilitato.

env

Facoltativo. L'ambiente di compilazione sostituisce l'attività. Questo può contenere le seguenti proprietà:

tipo di calcolo

L'identificatore del tipo di calcolo da utilizzare per l'attività. Vedere ComputeType [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) in per i valori possibili.

flotta

L'identificatore della flotta da utilizzare per l'attività. Per ulteriori informazioni, consulta [the section called “Run si basa su flotte a capacità riservata”](#).

image

L'identificatore dell'immagine da utilizzare per l'attività. Vedi Identificatore di immagine in [the section called “Immagini Docker fornite da CodeBuild”](#) per i valori possibili.

modalità privilegiata

Un valore booleano che indica se eseguire il demone Docker all'interno di un contenitore Docker. Imposta su true solo se il progetto di compilazione viene utilizzato per creare immagini Docker. In caso contrario, una build che tenta di interagire con il daemon Docker fallisce. L'impostazione predefinita è false.

tipo

L'identificatore del tipo di ambiente da utilizzare per l'attività. Vedi Environment type in [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) per i valori possibili.

variabili

Le variabili di ambiente che saranno presenti nell'ambiente di compilazione. Per ulteriori informazioni, consulta [env/variables](#).

Note

Tieni presente che il tipo di calcolo e il parco macchine non possono essere forniti nello stesso identificatore di una singola build.

ignore-failure

Facoltativo. Un valore booleano che indica se un errore di questa operazione di compilazione può essere ignorato.

false

Il valore predefinito. Se questa operazione di compilazione fallisce, la compilazione in batch avrà esito negativo.

true

Se questa operazione di compilazione fallisce, la compilazione in batch può comunque avere successo.

Di seguito è riportato un esempio di voce buildspec dell'elenco di compilazione:

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
```

```
buildspec: build2.yml
env:
  variables:
    BUILD_ID: build2
ignore-failure: true
- identifier: build3
  env:
    compute-type: ARM_LAMBDA_1GB
- identifier: build4
  env:
    fleet: fleet_name
- identifier: build5
  env:
    compute-type: GENERAL_LINUX_XLAGRE
```

batch/build-matrix

Definisce una matrice di compilazione. Una matrice di compilazione definisce le attività con diverse configurazioni eseguite in parallelo. CodeBuild crea una build separata per ogni possibile combinazione di configurazione. Per ulteriori informazioni, consulta [Costruisci una matrice](#).

statico

Le proprietà statiche si applicano a tutte le attività di compilazione.

ignore-failure

Facoltativo. Un valore booleano che indica se un errore di questa operazione di compilazione può essere ignorato.

false

Il valore predefinito. Se questa operazione di compilazione fallisce, la compilazione in batch avrà esito negativo.

true

Se questa operazione di compilazione fallisce, la compilazione in batch può comunque avere successo.

env

Facoltativo. L'ambiente di compilazione ha la precedenza su tutte le attività.

modalità privilegiata

Un valore booleano che indica se eseguire il demone Docker all'interno di un contenitore Docker. Imposta su `true` solo se il progetto di compilazione viene utilizzato per creare immagini Docker. In caso contrario, una build che tenta di interagire con il daemon Docker fallisce. L'impostazione predefinita è `false`.

tipo

L'identificatore del tipo di ambiente da utilizzare per l'attività. Vedi Environment type in [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) per i valori possibili.

dinamico

Le proprietà dinamiche definiscono la matrice di costruzione.

buildspec

Facoltativo. Un array che contiene il percorso e i nomi dei file buildspec da utilizzare per queste attività. Se questo parametro non è specificato, viene utilizzato il file buildspec corrente.

env

Facoltativo. L'ambiente di compilazione ha la precedenza su queste attività.

tipo di calcolo

Un array che contiene gli identificatori dei tipi di calcolo da utilizzare per queste attività. Vedere ComputeType [the section called “Modi e tipi di calcolo dell'ambiente di creazione”](#) in per i valori possibili.

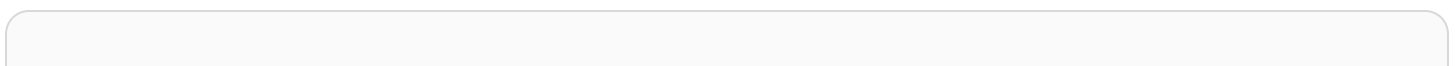
image

Un array che contiene gli identificatori delle immagini da utilizzare per queste attività. Vedi Image identifier in [the section called “Immagini Docker fornite da CodeBuild”](#) per i possibili valori.

variabili

Un array che contiene le variabili di ambiente che saranno presenti negli ambienti di compilazione per queste attività. Per ulteriori informazioni, consulta [env/variables](#).

Di seguito è riportato un esempio di una voce buildspec della matrice di compilazione:



```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Per ulteriori informazioni, consulta [Costruisci una matrice](#).

batch/build-fanout

Definisce un fanout di build. Un build fanout viene utilizzato per definire un'attività suddivisa in più build eseguite in parallelo. Per ulteriori informazioni, consulta [Esegui test paralleli in build in batch](#).

Questo elemento contiene un'attività di compilazione che può essere suddivisa in più build. La `build-fanout` sezione contiene le seguenti proprietà.

parallelismo

Obbligatorio. Il numero di build che eseguiranno i test in parallelo.

ignore-failure

Facoltativo. Un valore booleano che indica se un errore in una qualsiasi delle attività di compilazione di fanout può essere ignorato. Questo valore di `ignore-failure` verrà applicato a tutte le build di fanout.

false

Il valore predefinito. Se un'operazione di compilazione di fanout fallisce, la compilazione in batch avrà esito negativo.

true

Se un'operazione di compilazione di fanout fallisce, la compilazione in batch può comunque avere successo.

Di seguito è riportato un esempio di una voce build fanout buildspec:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'equal-distribution'
```

Per ulteriori informazioni, consulta [Crea fanout](#) e [Usa il comando codebuild-tests-run CLI](#).

Crea un riferimento ambientale per AWS CodeBuild

Quando AWS CodeBuild chiami per eseguire una build, devi fornire informazioni sull'ambiente di compilazione. Un ambiente di compilazione rappresenta una combinazione di sistema operativo, linguaggio di programmazione, runtime e strumenti CodeBuild utilizzati per eseguire una build. Per informazioni su come funziona un ambiente di compilazione, consulta [Come CodeBuild funziona](#).

Un ambiente di compilazione contiene un'immagine Docker. Per informazioni, consulta il [glossario Docker](#) sul sito Web Docker Docs.

Quando fornisci informazioni CodeBuild sull'ambiente di compilazione, specifichi l'identificatore di un'immagine Docker in un tipo di repository supportato. Questi includono l'archivio di immagini CodeBuild Docker, le immagini disponibili pubblicamente in Docker Hub e gli archivi Amazon Elastic Container Registry (Amazon ECR) a cui il tuo account dispone delle autorizzazioni di accesso. AWS

- Ti consigliamo di utilizzare immagini Docker archiviate nell'archivio di immagini CodeBuild Docker, poiché sono ottimizzate per l'uso con il servizio. Per ulteriori informazioni, consulta [Immagini Docker fornite da CodeBuild](#).
- Per ottenere l'identificatore di un'immagine Docker disponibile pubblicamente archiviata in Docker Hub, consulta [Ricerca di repository](#) sul sito Web Docker Docs.
- Per informazioni su come utilizzare le immagini Docker archiviate nei repository Amazon ECR del tuo AWS account, consulta [Esempio di Amazon ECR](#).

Oltre a un identificatore dell'immagine Docker, è anche possibile specificare un insieme di risorse di elaborazione che l'ambiente di compilazione utilizza. Per ulteriori informazioni, consulta [Modi e tipi di calcolo dell'ambiente di creazione](#).

Argomenti

- [Immagini Docker fornite da CodeBuild](#)
- [Modi e tipi di calcolo dell'ambiente di creazione](#)
- [Shell e comandi negli ambienti di compilazione](#)
- [Variabili di ambiente degli ambienti di compilazione](#)
- [Attività in background degli ambienti di compilazione](#)

Immagini Docker fornite da CodeBuild

Un'immagine supportata è l'ultima versione principale di un'immagine disponibile in CodeBuild e viene aggiornata con aggiornamenti secondari e patch. CodeBuild ottimizza la durata del provisioning delle build con immagini supportate memorizzandole nella cache nell'[Amazon Machine Images \(AMI\)](#) della macchina. Se desideri trarre vantaggio dalla memorizzazione nella cache e ridurre al minimo la durata del provisioning della build, seleziona Usa sempre l'immagine più recente per questa versione di runtime nella sezione Versione immagine della CodeBuild console anziché una versione più granulare, ad esempio. `aws/codebuild/amazonlinux-x86_64-standard:4.0-1.0.0`

Argomenti

- [Ottieni l'elenco delle immagini Docker correnti](#)
- [EC2 calcola immagini](#)
- [Immagini di calcolo Lambda](#)
- [Immagini obsolete CodeBuild](#)
- [Runtime disponibili](#)
- [Versioni di runtime](#)

Ottieni l'elenco delle immagini Docker correnti

CodeBuild aggiorna frequentemente l'elenco delle immagini Docker per aggiungere le immagini più recenti e rendere obsolete le vecchie immagini. Per ottenere l'elenco più aggiornato, procedere in uno dei seguenti modi:

- Nella CodeBuild console, nella procedura guidata Crea progetto di compilazione o nella pagina Modifica progetto di compilazione, per Immagine ambientale, scegli Immagine gestita. Scegliere tra gli elenchi a discesa Operating system (Sistema operativo), Runtime (Runtime) e Runtime version (Versione runtime). Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#) o [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).
- Per la AWS CLI, esegui il `list-curated-environment-images` comando:

```
aws codebuild list-curated-environment-images
```

- Per il AWS SDKs, chiamate l'`ListCuratedEnvironmentImages` operazione per il linguaggio di programmazione di destinazione. Per ulteriori informazioni, consulta [AWS SDKs e riferimento agli strumenti](#).

EC2 calcola immagini

AWS CodeBuild supporta le seguenti immagini Docker disponibili per il EC2 calcolo in. CodeBuild

Note

L'immagine di base della piattaforma Windows Server Core 2019 è disponibile solo nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (Oregon)
- Europa (Irlanda)

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto8	al/standard/corretto8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:corretto11	al/standard/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D
Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D

Note

Il 22 novembre 2024, gli alias per le immagini di runtime standard basate su Linux sono stati aggiornati da `al` a `amazonlinux2` `amazonlinux`. Non è richiesto alcun aggiornamento manuale poiché gli alias precedenti sono ancora validi.

Immagini di calcolo Lambda

AWS CodeBuild supporta le seguenti immagini Docker disponibili per il AWS Lambda calcolo. CodeBuild

⚠ Important

L'on-failureattributo non è supportato quando si utilizza l'elaborazione Lambda o la capacità riservata. Questo attributo funziona solo con le immagini di EC2 calcolo fornite da CodeBuild

Architettura di aarch64

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6	al-6-lambda/aarch64/dotnet
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8	al-lambda/aarch64/dotnet 8
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21	al-lambda/aarch64/go 1.21
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.24	al-1,24-lambda/aarch64/go
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:correcto11	al-11-lambda/aarch64/corretto
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:correcto17	al-lambda/aarch64/corretto 17

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
	<code>mbda-standard:corretto17</code>	
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21</code>	alle lambda/aarch64/corretto 21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18</code>	al lambda/aarch64/nodejs 18
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20</code>	al lambda/aarch64/nodejs 20
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs22</code>	al lambda/aarch64/nodejs 22
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11</code>	al- lambda/aarch64/python 3.11
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12</code>	al- 3.12 lambda/aarch64/python hon

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.13	al-3.13 lambda/aarch64/python
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2	al-3,2 lambda/aarch64/ruby
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.4	lambda/aarch64/rubyal-3,4

Architettura di **x86_64**

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6	al- lambda/x86_64/dotnet 6
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8	al- lambda/x86_64/dotnet 8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21	al- lambda/x86_64/go 1.21

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.24	al-1,24 lambda/x86_64/go
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11	al-11 lambda/x86_64/corretto
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17	al lambda/x86_64/corretto 17
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21	alle lambda/x86_64/corretto 21
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18	al lambda/x86_64/nodejs 18
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20	al lambda/x86_64/nodejs 20
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs22	al lambda/x86_64/nodejs 22

Platform (Piattaforma)	Identificatore dell'immagine	Definizione
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11	al-lambda/x86_64/python 3.11
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12	al- 3.12 lambda/x86_64/python
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.13	al- 3.13 lambda/x86_64/python
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2	al-3,2 lambda/x86_64/ruby
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.4	lambda/x86_64/rubyal-3,4

Immagini obsolete CodeBuild

Un'immagine obsoleta è un'immagine che non viene più memorizzata nella cache o aggiornata da CodeBuild. Un'immagine obsoleta non riceve più aggiornamenti di versione secondari o aggiornamenti delle versioni delle patch e, poiché non sono più aggiornate, il loro utilizzo potrebbe non essere sicuro. Se il CodeBuild progetto è configurato per utilizzare una versione di immagine precedente, il processo di provisioning scaricherà questa immagine docker e la utilizzerà per creare l'ambiente di runtime containerizzato, che può aumentare la durata del provisioning e la durata complessiva della build.

CodeBuild ha reso obsolete le seguenti immagini Docker. Puoi comunque utilizzare queste immagini, ma non verranno memorizzate nella cache dell'host di build e comporteranno tempi di provisioning più lunghi.

Platform (Piattaforma)	Identificatore dell'immagine	Definizione	Data di ritiro
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:3.0	al2/standard/3.0	9 maggio 2023
Ubuntu 18.04	aws/codebuild/ standard:4.0	ubuntu/standard/4.0	31 marzo 2023
Amazon Linux 2	aws/codebuild/ amazonlinux2- aarch64-s tandard:1.0	al2/aarch64/standa rd/1.0	31 marzo 2023
Ubuntu 18.04	aws/codebuild/ standard:3.0	ubuntu/standard/3.0	30 giugno 2022
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:2.0	al2/standard/2.0	30 giugno 2022

Argomenti

- [Runtime disponibili](#)
- [Versioni di runtime](#)

Runtime disponibili

È possibile specificare uno o più runtime nella sezione `runtime-versions` del file `buildspec`. Se il runtime dipende da un altro runtime, è anche possibile specificarne il runtime dipendente

nel file `buildspec`. Se non specificate alcun runtime nel file `buildspec`, CodeBuild sceglie i runtime predefiniti disponibili nell'immagine che utilizzate. Se specificate uno o più runtime, utilizza solo tali runtime. CodeBuild Se non viene specificato un runtime dipendente, CodeBuild tenta di scegliere automaticamente il runtime dipendente. Per ulteriori informazioni, consulta [Specify runtime versions in the buildspec file](#).

Argomenti

- [Runtime di immagini Linux](#)
- [Runtime delle immagini macOS](#)
- [Runtime di esecuzione delle immagini Windows](#)

Runtime di immagini Linux

La tabella seguente contiene i runtime disponibili e le immagini Linux standard che li supportano.

Runtime delle piattaforme Ubuntu e Amazon Linux

Nome runtime	Versione	Immagini
dotnet	6.0	Amazon Linux 2 x86_64 Standard Lambda: dotnet6
		Standard Amazon Linux 2 AArch64 Lambda: dotnet6
		Standard Amazon Linux 2023 x86_64:5.0
		AArch64Standard Amazon Linux 2023:3.0
	8.0	Ubuntu standard: 7.0
		Standard Amazon Linux 2023 x86_64:5.0
		AArch64Standard Amazon Linux 2023:3.0

Nome runtime	Versione	Immagini
golang		Ubuntu standard: 7.0
	1.20	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	1.21	Amazon Linux 2 x86_64 Standard Lambda: go1.21 Standard Amazon Linux 2 AArch64 Lambda: go1.21 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	1.22	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	1.23	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	1.24	Amazon Linux 2023 x86_64 Standard Lambda: go1.24 Standard Amazon Linux 2023 AArch64 Lambda: go1.24 Standard Amazon Linux 2023 x86_64:5.0 Ubuntu standard: 7.0
java	corretto8	Amazon Linux 2 x86_64 standard:corretto8 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	corretto11	Amazon Linux 2 x86_64 standard:corretto11 Amazon Linux 2 x86_64 Lambda Standard: corretto11 Standard Amazon Linux 2023 x86_64:5.0 Standard Amazon Linux 2 AArch64 Lambda: corretto11 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	corretto 17	Amazon Linux 2 x86_64 Lambda Standard: corretto17 Standard Amazon Linux 2 AArch64 Lambda: corretto17 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	corretto 21	Amazon Linux 2 x86_64 Lambda Standard: corretto21 Standard Amazon Linux 2 AArch64 Lambda: corretto21 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	corretto 25	AArch64Standard Amazon Linux 2023:3.0

Nome runtime	Versione	Immagini
nodejs	18	Standard Lambda Amazon Linux 2 x86_64: nodejs18 Standard Amazon Linux 2 AArch64 Lambda: nodejs18 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	20	Standard Lambda Amazon Linux 2 x86_64: nodejs20 Standard Amazon Linux 2 AArch64 Lambda: nodejs20 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	22	Amazon Linux 2023 x86_64 Standard Lambda: nodejs22 Standard Amazon Linux 2023 AArch64 Lambda: nodejs22 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	24	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
php	8.1	AArch64Standard Amazon Linux 2023:3.0
	8.2	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	8.3	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	8.4	AArch64Standard Amazon Linux 2023:3.0
	8.5	AArch64Standard Amazon Linux 2023:3.0
python	3.9	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	3.10	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	3.11	Standard Lambda Amazon Linux 2 x86_64: python3.11 Standard Amazon Linux 2 AArch64 Lambda: python3.11 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	3.12	Amazon Linux 2 x86_64 Standard Lambda: python3.12 Standard Amazon Linux 2 AArch64 Lambda: python3.12 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	3.13	Amazon Linux 2023 x86_64 Standard Lambda: python3.13 Standard Amazon Linux 2023 AArch64 Lambda: python3.13 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	3.14	AArch64Standard Amazon Linux 2023:3.0
ruby	3.1	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	3.2	Amazon Linux 2 x86_64 Standard Lambda: ruby3.2 Standard Amazon Linux 2 AArch64 Lambda: ruby3.2 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Nome runtime	Versione	Immagini
	3.3	Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0
	3.4	Amazon Linux 2023 x86_64 Standard Lambda: ruby3.4 Standard Amazon Linux 2023 AArch64 Lambda: ruby3.4 Standard Amazon Linux 2023 x86_64:5.0 AArch64Standard Amazon Linux 2023:3.0 Ubuntu standard: 7.0

Runtime delle immagini macOS



Important

Le immagini CodeBuild curate per le build Mac contengono macOS e Xcode preinstallati. [Utilizzando il software Xcode, riconosci, comprendi e acconsenti al Contratto Xcode e Apple SDKs](#) Se non accettate i termini e le condizioni del contratto, non utilizzate il software Xcode. Fornisci invece le tue Amazon Machine Images (AMI). Per ulteriori informazioni, consulta [Come posso configurare una flotta macOS a capacità riservata?](#)

La tabella seguente contiene i runtime disponibili supportati da macOS.

Runtime della piattaforma macOS

Nome runtime	Versione	Immagini	Note aggiuntive
bash	3.2.57	macos-arm-base3,2,57 ----set----:15 macos-arm-base:15 ----set----:26	
clang	17.0.0	macos-arm-base17.0.0 ----set----:15 macos-arm-base:15 ----set----:26	
dotnet sdk	8.0.416	macos-arm-base8,0,416 ----set----:15	
	8,0,417	macos-arm-base8,0,417 ----set----:26	
	10,0,101	macos-arm-base10,0101 ----set----:15	
	10,0,102	macos-arm-base10,0102 ----set----:26	
gcc	11,5,0	macos-arm-base11.5.0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gcc-11
	12.4.0	macos-arm-base12,4.0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gcc-12

Nome runtime	Versione	Immagini	Note aggiuntive
	13.4.0	macos-arm-base13,4 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gcc-13
	14.3.0	macos-arm-base14.3 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gcc-14
gnu	11.5.0	macos-arm-base11.5 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gfortran-11
	12.4.0	macos-arm-base12,4 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gfortran-12
	13.4.0	macos-arm-base13,4 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gfortran-13
	14.3.0	macos-arm-base14.3 .0 ----set----:15 macos-arm-base:15 ----set----:26	Disponibile utilizzando l'alias gfortran-14

Nome runtime	Versione	Immagini	Note aggiuntive
golang	1.24.11	macos-arm-base1,24 .11 ----set----:15 macos-arm-base:15 ----set----:26	
	1.25.4	macos-arm-base:26 ----set----:15 macos-arm-base:15 ----set----:26	
java	Corretto 8	macos-arm-baseCorr etto 8 ----- 15 macos-arm-base:15 ----set----:26	
	Corretto 11	macos-arm-baseCorr etto 11 ----- 15 macos-arm-base:15 ----set----:26	
	Corretto 17	macos-arm-baseCorr etto 17 ----- 15 macos-arm-base:15 ----set----:26	
	Corretto 21	macos-arm-baseCorr etto 21 ----- 15 macos-arm-base:15 ----set----:26	

Nome runtime	Versione	Immagini	Note aggiuntive
	Corretto 25	macos-arm-baseCorretto 25 ----- 15 macos-arm-base:15 ----set----:26	
kotlin	2.2.21	macos-arm-base2,2,21 ----set----:15	
	2.3.0	macos-arm-base:15 ----set----:26	
mono	6.14.1	macos-arm-base6.14.1 ----set----:15 macos-arm-base:15 ----set----:26	
nodejs	20,19,6	macos-arm-base20,19,6 ----set----:15 macos-arm-base:15 ----set----:26	
	22,21,1	macos-arm-base22.21.1 ----set----:15 macos-arm-base:15 ----set----:26	
	24,11,1	macos-arm-base24.11.1 ----set----:15 macos-arm-base:15 ----set----:26	

Nome runtime	Versione	Immagini	Note aggiuntive
perl	5,34,1	macos-arm-base5,34,1 ----set----:15 macos-arm-base:15 ----set----:26	
php	8,2,29	macos-arm-base8,2,29 ----set----:15	
	8,2,30	macos-arm-base8,2,30 ----set----:26	
	8,3,28	macos-arm-base8,3,28 ----set----:15	
	8,3,30	macos-arm-base8,3,30 ----set----:26	
	8,4,15	macos-arm-base8,4,15 ----set----:15	
	8,4,17	macos-arm-base8,4,17 ----set----:26	
	8,5,0	macos-arm-base8.5.0 ----set----:15	
	8.5.2	macos-arm-base8.5.2 ----set----:26	
python	3,10,19	macos-arm-base3,10,19 ----set----:15 macos-arm-base:15 ----set----:26	

Nome runtime	Versione	Immagini	Note aggiuntive
	3,11,14	macos-arm-base3,11,14 ----set----:15 macos-arm-base:15 ----set----:26	
	3,12,12	macos-arm-base3,12,12 ----set----:15 macos-arm-base:15 ----set----:26	
	3,13,10	macos-arm-base3,13,10 ----set----:15 macos-arm-base:15 ----set----:26	
	3,14,1	macos-arm-base3.14.1 ----set----:15 macos-arm-base:15 ----set----:26	
ruby	3.2,9	macos-arm-base3.2.9 ----set----:15 macos-arm-base:15 ----set----:26	
	3,3,10	macos-arm-base3,3,10 ----set----:15 macos-arm-base:15 ----set----:26	

Nome runtime	Versione	Immagini	Note aggiuntive
	3,4,7	macos-arm-base3.4.7 ----set----:15 macos-arm-base:15 ----set----:26	
rust	1.91.1	macos-arm-base1,91 ,1 ----set----:15	
	1,92,0	macos-arm-base1,92 .0 ----set----:26	
veloce	6.2.1	macos-arm-base6.2.1 ----set----:15	
	6.2.3	macos-arm-base6.2.3 ----set----:26	
Xcode	26.1.1	macos-arm-baseXcod e ----sep----:15	
	26,2	macos-arm-base26,2 ----set----:26	

Runtime di esecuzione delle immagini Windows

L'immagine di base di Windows Server Core 2019 contiene i seguenti runtime.

Runtime della piattaforma Windows

Nome runtime	Windows Server Core 2019 standard: versioni 1.0	Windows Server Core 2019 standard:versioni 2.0	Windows Server Core 2019 standard:versioni 3.0
dotnet	3.1	3.1	8.0
	5.0	6.0	

Nome runtime	Windows Server Core 2019 standard: versioni 1.0	Windows Server Core 2019 standard:versioni 2.0	Windows Server Core 2019 standard:versioni 3.0
		7.0	
dotnet sdk	3.1 5.0	3.1 6.0 7.0	8.0
golang	1.14	1.18	1,21 1,22 1,23
gradino	6.7	7.6	8.12
java	Corretto 11	Corretto 11 Corretto 17	Corretto 8 Corretto 11 Corretto 17 Corretto 21
esperta	3.6	3.8	3.9
nodejs	14.15	16,19	20,18 22,13
php	7.4	8.1	8.3 8.4
powershell	7.1	7.2	7.4

Nome runtime	Windows Server Core 2019 standard: versioni 1.0	Windows Server Core 2019 standard:versioni 2.0	Windows Server Core 2019 standard:versioni 3.0
python	3.8	3.10	3,10
			3,11
			3,12
			3.13
ruby	2.7	3.1	3.2
			3.3
			3.4

Versioni di runtime

Quando si specifica un runtime nella sezione [runtime-versions](#) del file buildspec, è possibile specificare una versione specifica, una versione principale specifica e l'ultima versione secondaria o la versione più recente. Nella tabella seguente sono elencati i runtime disponibili e come specificarli. Non tutte le versioni di runtime sono disponibili su tutte le immagini. Inoltre, la selezione della versione di runtime non è supportata per le immagini personalizzate. Per ulteriori informazioni, consulta [Runtime disponibili](#). Se desideri installare e utilizzare una versione di runtime personalizzata anziché le versioni di runtime preinstallate, consulta [Versioni di runtime personalizzate](#).

Versioni di runtime della piattaforma Ubuntu e Amazon Linux 2

Nome runtime	Versione	Versione specifica	Versione principale e secondari a specifica	Versione più recente
android	28	android: 28	android: 28.x	android: latest
	29	android: 29	android: 29.x	

Nome runtime	Versione	Versione specifica	Versione principale e secondari a specifica	Versione più recente
dotnet	3.1	dotnet: 3.1	dotnet: 3.x	dotnet: latest
	5.0	dotnet: 5.0	dotnet: 5.x	
	6.0	dotnet: 6.0	dotnet: 6.x	
	8.0	dotnet: 8.0	dotnet: 8.x	
golang	1.12	golang: 1.12	golang: 1.x	golang: latest
	1.13	golang: 1.13		
	1.14	golang: 1.14		
	1.15	golang: 1.15		
	1.16	golang: 1.16		
	1.18	golang: 1.18		
	1,20	golang: 1.20		
	1,21	golang: 1.21		
	1,22	golang: 1.22		
	1,23	golang: 1.23		
	1,24	golang: 1.24		
java	corretto8	java: corretto	java: corretto .x	java: latest
	corretto11	java: corretto 1	java: corretto 1.x	

Nome runtime	Versione	Versione specifica	Versione principale e secondari a specifica	Versione più recente
	corretto 17	java: corretto 7	java: corretto 7.x	
	corretto 21	java: corretto 1	java: corretto 1.x	
nodejs	10	nodejs: 10	nodejs: 10.x	nodejs: latest
	12	nodejs: 12	nodejs: 12.x	
	14	nodejs: 14	nodejs: 14.x	
	16	nodejs: 16	nodejs: 16.x	
	18	nodejs: 18	nodejs: 18.x	
	20	nodejs: 20	nodejs: 20.x	
	22	nodejs: 22	nodejs: 22.x	
php	7.3	php: 7.3	php: 7.x	php: latest
	7.4	php: 7.4	php: 8.x	
	8.0	php: 8.0		
	8.1	php: 8.1		
	8.2	php: 8.2		
	8.3	php: 8.3		
python	3.7	python: 3.7	python: 3.x	python: latest
	3.8	python: 3.8		

Nome runtime	Versione	Versione specifica	Versione principale e secondari a specifica	Versione più recente
	3.9	python: 3.9		
	3,10	python: 3.10		
	3,11	python: 3.11		
	3,12	python: 3.12		
	3.13	python: 3.13		
ruby	2.6	ruby: 2.6	ruby: 2.x	ruby: latest
	2.7	ruby: 2.7	ruby: 3.x	
	3.1	ruby: 3.1		
	3.2	ruby: 3.2		
	3.3	ruby: 3.3		
	3.4	ruby: 3.4		

È possibile utilizzare una specifica di build per installare altri componenti (ad esempio, Apache Maven AWS CLI, Apache Ant, Mocha o simili) durante la RSpec fase di compilazione. `install` Per ulteriori informazioni, consulta [Esempio di buildspec](#).

Versioni di runtime personalizzate

Invece di utilizzare le versioni di runtime preinstallate nelle immagini CodeBuild gestite, puoi installare e utilizzare versioni personalizzate a tua scelta. La tabella seguente elenca i runtime personalizzati disponibili e come specificarli.

 Note

La selezione personalizzata della versione di runtime è supportata solo per le immagini di Ubuntu e Amazon Linux.

Versioni di runtime personalizzate

Nome runtime	Sintassi	Esempio
dotnet	<i><major>.<minor>.<patch></i>	5.0.408
golang	<i><major>.<minor></i>	1.19
	<i><major>.<minor>.<patch></i>	1.19.1
java	corretto <i><major></i>	corretto15
nodejs	<i><major></i>	14
	<i><major>.<minor></i>	14.21
	<i><major>.<minor>.<patch></i>	14.21.3
php	<i><major>.<minor>.<patch></i>	8.0.30
python	<i><major></i>	3
	<i><major>.<minor></i>	3.7
	<i><major>.<minor>.<patch></i>	3.7.16
ruby	<i><major>.<minor>.<patch></i>	3.0.6

Esempio di buildspec di runtime personalizzato

Ecco un esempio di buildspec che specifica versioni di runtime personalizzate.

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

```
runtime-versions:
  java: corretto15
  php: 8.0.30
  ruby: 3.0.6
  golang: 1.19
  python: 3.7
  nodejs: 14
  dotnet: 5.0.408
```

Modi e tipi di calcolo dell'ambiente di creazione

In CodeBuild, puoi specificare l'immagine dell'ambiente di calcolo e di runtime da CodeBuild utilizzare per eseguire le build. Compute si riferisce al motore di elaborazione (CPU, memoria e sistema operativo) gestito e mantenuto da CodeBuild. Un'immagine dell'ambiente di runtime è un'immagine contenitore che viene eseguita sulla piattaforma di elaborazione scelta e include strumenti aggiuntivi di cui la build potrebbe aver bisogno, come AWS CLI.

Argomenti

- [Informazioni sull'elaborazione](#)
- [Informazioni sui tipi di ambienti con capacità riservata](#)
- [Informazioni sui tipi di ambiente on-demand](#)

Informazioni sull'elaborazione

CodeBuild offre EC2 e modalità di AWS Lambda calcolo. EC2 offre una flessibilità ottimizzata durante la costruzione e AWS Lambda offre velocità di avvio ottimizzate. AWS Lambda supporta build più veloci grazie a una minore latenza di avvio. AWS Lambda inoltre si ridimensiona automaticamente, quindi le build non attendono in coda per essere eseguite. Per ulteriori informazioni, consulta [Esegui build su calcolo AWS Lambda](#).

Con la modalità di EC2 elaborazione, puoi eseguire le tue build con flotte su richiesta o con capacità riservata. Per le flotte on-demand, puoi selezionare tipi di elaborazione predefiniti come o. BUILD_GENERAL1_SMALL BUILD_GENERAL1_LARGE Per ulteriori informazioni, consulta [Informazioni sui tipi di ambiente on-demand](#). Per le flotte a capacità riservata, puoi selezionare le configurazioni di elaborazione tra cui vCPU, memoria e spazio su disco. Dopo aver specificato le configurazioni, CodeBuild sceglierà un tipo di elaborazione supportato che soddisfi i tuoi requisiti. Per ulteriori informazioni, consulta [Informazioni sui tipi di ambienti con capacità riservata](#).

Informazioni sui tipi di ambienti con capacità riservata

AWS CodeBuild fornisce tipi di ambiente Linux x86, Arm, GPU, Windows e macOS per flotte a capacità riservata. La tabella seguente mostra il tipo di macchina, la memoria, v e lo spazio su disco disponibili CPUs, ordinati per regione:

US East (N. Virginia)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu. .128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
GPU Linux	96	384 GiB	3785 GB (SSD)	NVME	reserved.gpu.96cpu.384gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

US East (Ohio)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

US West (Oregon)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
GPU Linux	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
macOS	12	32 GiB	256 GB	GENERAL	reserved. arm.m2.12 cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Tokyo)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Mumbai)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Singapore)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Asia Pacific (Sydney)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	16	32 GiB	256 GB	GENERAL	reserved. arm.16cpu .32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved. arm.32cpu .64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved. arm.48cpu .96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu .128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Europe (Frankfurt)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved. arm.16cpu. .32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved. arm.32cpu. .64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved. arm.48cpu. .96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu. .128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved. gpu.4cpu. 16gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gi b
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Europe (Ireland)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved. arm.16cpu .32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved. arm.32cpu .64gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	48	96 GiB	512 GB	GENERAL	reserved. arm.48cpu. .96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved. arm.64cpu. .128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved. arm.4cpu. 8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved. arm.8cpu. 16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved. x86-64.48 cpu.96gib .nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved. x86-64.72 cpu.144gib .nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
GPU Linux	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved. x86-64.96 cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

South America (São Paulo)

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	2	4 GiB	64 GB	GENERAL	reserved. arm.2cpu. 4gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
BRACCIO EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
BRACCIO EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
BRACCIO EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	2	4 GiB	64 GB	GENERAL	reserved. x86-64.2c pu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved. x86-64.16 cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gi b

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipo di ambiente	v CPUs	Memoria	Spazio su disco	Tipo di macchina	Tipo di istanza di calcolo
Windows	36	72 GiB	256 GB	GENERAL	reserved. x86-64.36 cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved. x86-64.48 cpu.96gib
Windows	72	144 GiB	824 GB	GENERAL	reserved. x86-64.72 cpu.144gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved. x86-64.4c pu.8gib
Finestre EC2	8	16 GiB	128 GB	GENERAL	reserved. x86-64.8c pu.16gib

Per ulteriori informazioni sull'identificativo del prezzo, consulta <https://aws.amazon.com/codebuild/pricing/>.

Per scegliere un tipo di calcolo:

- Nella CodeBuild console, nella pagina di configurazione di Compute fleet, scegli una delle opzioni tra v CPUs, Memoria e Disco. Per ulteriori informazioni, consulta [Crea un parco veicoli a capacità riservata](#).

- Per la AWS CLI, esegui il `update-fleet` comando `create-fleet` o, specificando il valore di `computeType` `to. ATTRIBUTE_BASED_COMPUTE` [Per ulteriori informazioni, consulta `create-fleet` o `update-fleet`.](#)
- Per la AWS SDKs, chiamate l'equivalente dell'UpdateFleetoperazione `CreateFleet` o per il linguaggio di programmazione di destinazione, specificando il valore di `to. computeType` `ATTRIBUTE_BASED_COMPUTE` Per ulteriori informazioni, consulta [AWS SDKs e riferimento agli strumenti.](#)

Note

Per la AWS CLI e AWS SDKs, puoi comunque utilizzare `computeType` input come `BUILD_GENERAL1_SMALL`, per scegliere i tipi di calcolo anziché.

`ATTRIBUTE_BASED_COMPUTE` Per ulteriori informazioni, consulta [Informazioni sui tipi di ambiente on-demand.](#)

Famiglie di istanza supportate

AWS CodeBuild supporta le seguenti istanze per flotte a capacità riservata. :

- Uso generico: M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8g | T3 | T3a | T4g
- Ottimizzate per il calcolo: C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8g
- Memoria ottimizzata: R5 | R5a | R5ad | R5b | R5d | R5dn | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7gd | R7i | R7iZ | R8g | U-3TB1 | U-6TB1 | U-9TB1 | U-12TB1 | U-18TB1 | U-24TB1 | U7i-6TB | U7i-12TB | U7in-16TB | U7in-24TB | U7in-32TB | X1 | X1e | X2gD | X2idn | X2iEDN | X2iEZn | X8g | z1d
- Archiviazione ottimizzata: D3 | D3en | I3 | I3en | i4G | i4i | i7le | i8G | Im4gn | IS4Gen
- Calcolo accelerato: DL1 | DL2q | F1 | F2 | G4ad | G4dn | G5 | G5 | G6 | G6e | Gr6 | Inf1 | Inf2 | P3 | P3dn | P4d | P5 | P5e | P5en | Trn1 | Trn1n | Trn2 | VT1
- Calcolo ad alte prestazioni: Hpc6a | Hpc6id | Hpc7a | Hpc7g
- Generazione precedente: A1

Per creare una flotta di capacità riservata con un tipo di istanza specifico:

- Nella CodeBuild console, nella pagina di configurazione della flotta Compute, vai alla sezione Configurazione della capacità. Nella modalità di selezione Compute, seleziona Input manuale e in Tipo di istanza Compute scegli uno dei tipi di istanza dal menu a discesa. Per ulteriori informazioni, consulta [Crea un parco veicoli a capacità riservata](#).
- Per la AWS CLI, esegui il `update-fleet` comando `create-fleet` or, specificando il valore di `computeType` to `CUSTOM_INSTANCE_TYPE` e il `ComputeConfiguration instanceType` tipo di istanza specificato. [Per ulteriori informazioni, consulta `create-fleet` o `update-fleet`](#).
- Per la AWS SDKs, chiamate l'equivalente dell'UpdateFleetoperazione `CreateFleet` or per il linguaggio di programmazione di destinazione, specificando il valore di `computeType` to `CUSTOM_INSTANCE_TYPE` e il `ComputeConfigurationinstanceType` tipo di istanza specificato. Per ulteriori informazioni, consulta [AWS SDKs e riferimento agli strumenti](#).

Informazioni sui tipi di ambiente on-demand

AWS CodeBuild fornisce agli ambienti di compilazione la seguente memoria CPUs, v e spazio su disco disponibili per la modalità di EC2 elaborazione:

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	v CPUs	Spazio su disco
ARM Piccolo ¹	BUILD_GENERAL1_SMALL	ARM_CONTAINER ARM_EC2	4 GiB	2	64 GB
ARM Medio ¹	BUILD_GENERAL1_MEDIUM	ARM_CONTAINER ARM_EC2	8 GiB	4	128 GB
ARM Large ¹	BUILD_GENERAL1_LARGE	ARM_CONTAINER ARM_EC2	16 GiB	8	128 GB

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	v CPUs	Spazio su disco
XLarge BRACCIO ¹	BUILD_GENERAL1_XLARGE	ARM_CONTAINER	64 GiB	32	256 GB
BRACCIO XLarge 2 ¹	BUILD_GENERAL1_2XLARGE	ARM_CONTAINER	96 GiB	48	824 GB
Linux Small ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER LINUX_EC2	4 GiB	2	64 GB
Linux Medium ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER LINUX_EC2	8 GiB	4	128 GB
Linux Large ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER LINUX_EC2	16 GiB	8	128 GB
Linux XLarge ¹	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	72 GiB	36	256 GB
Linux 2 XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	144 GiB	72	824 GB (SSD)
GPU Linux piccola	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GiB	4	235 GB (SSD)

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	v CPUs	Spazio su disco
GPU Linux di grandi dimensioni	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GiB	32	50 GB
Windows Medio ¹	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	8 GiB	4	128 GB
		WINDOWS_SERVER_2022_CONTAINER			
		WINDOWS_EC2			
Finestre di grandi dimensioni ¹	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER WINDOWS_SERVER_2022_CONTAINER WINDOWS_EC2	16 GiB	8	128 GB

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	v CPUs	Spazio su disco
Finestre XLarge ¹	BUILD_GENERAL1_XLARGE	WINDOWS_SERVER_2022_CONTAINER	72 GiB	36	256 GB
Windows 2 ¹ XLarge	BUILD_GENERAL1_2XLARGE	WINDOWS_SERVER_2022_CONTAINER	144 GiB	72	824 GB

¹ L'ultima versione di questo tipo di immagine è memorizzata nella cache. Se specifichi una versione più specifica, effettua il CodeBuild provisioning di quella versione anziché della versione memorizzata nella cache. Ciò può comportare tempi di build più lunghi. Ad esempio, per beneficiare della memorizzazione nella cache, specificare `aws/codebuild/amazonlinux-x86_64-standard:5.0` invece di una versione più granulare, ad esempio `aws/codebuild/amazonlinux-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild fornisce agli ambienti di compilazione la memoria e lo spazio su disco disponibili seguenti per la modalità di AWS Lambda elaborazione:

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	Spazio su disco
ARM Lambda 1 GB	BUILD_LAMBDA_1GB	ARM_LAMBDA_CONTAINER	1 GiB	10 GB
ARM Lambda 2 GB	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GiB	10 GB

Tipo di calcolo	Valore di Environment ComputeType	Valore del tipo di ambiente	Memoria	Spazio su disco
ARM Lambda 4 GB	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GiB	10 GB
ARM Lambda 8 GB	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GiB	10 GB
ARM Lambda 10 GB	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 GiB	10 GB
Linux Lambda 1 GB	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	1 GiB	10 GB
Linux Lambda 2 GB	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GiB	10 GB
Linux Lambda 4 GB	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GiB	10 GB
Linux Lambda da 8 GB	BUILD_LAMBDA_8GB	LINUX_LAMBDA_CONTAINER	8 GiB	10 GB
Linux Lambda 10 GB	BUILD_LAMBDA_10GB	LINUX_LAMBDA_CONTAINER	10 GiB	10 GB

Quando si utilizzano altri tipi di ambiente, si consiglia di utilizzare un'immagine memorizzata nella cache per ridurre i tempi di creazione.

Lo spazio su disco specifico per ciascun ambiente di compilazione è disponibile solo nella directory definita dalla variabile di ambiente `CODEBUILD_SRC_DIR`.

Per scegliere un tipo di calcolo:

- Nella CodeBuild console, nella procedura guidata Crea progetto di compilazione o nella pagina Modifica progetto di compilazione, in Ambiente espandi Configurazione aggiuntiva, quindi scegli una delle opzioni da Tipo di calcolo. Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#) o [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).
- Per la AWS CLI, esegui il `update-project` comando `create-project` or, specificando il `computeType` valore dell'oggetto. `environment` Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#) o [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#).
- Per la AWS SDKs, chiamate l'equivalente dell'UpdateProjectoperazione `CreateProject` o per il linguaggio di programmazione di destinazione, specificando l'equivalente del `computeType` valore dell'environmentoggetto. Per ulteriori informazioni, consulta [AWS SDKs e riferimento agli strumenti](#).

Alcuni tipi di ambiente e di elaborazione presentano limitazioni di disponibilità a livello regionale:

- Il tipo di elaborazione Linux GPU Small (`LINUX_GPU_CONTAINER`) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Virginia settentrionale)
 - US West (Oregon)
 - Asia Pacifico (Tokyo)
 - Canada (Centrale)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Europa (Londra)
- Il tipo di elaborazione Linux GPU Large (`LINUX_GPU_CONTAINER`) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - US West (Oregon)

- Asia Pacifico (Seoul)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Il tipo di elaborazione BUILD_GENERAL1_2XLARGE è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti occidentali (California settentrionale)
 - US West (Oregon)
 - Asia Pacific (Hyderabad)
 - Asia Pacifico (Hong Kong)
 - Asia Pacifico (Giacarta)
 - Asia Pacifico (Melbourne)
 - Asia Pacifico (Mumbai)
 - Asia Pacifico (Seoul)
 - Asia Pacifico (Singapore)
 - Asia Pacifico (Sydney)
 - Asia Pacifico (Tokyo)
 - Canada (Centrale)
 - Cina (Pechino)
 - China (Ningxia)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Europa (Londra)
 - Europa (Parigi)

- Europa (Spagna)
- Europa (Stoccolma)
- Europa (Zurigo)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- Il tipo di ambiente ARM_CONTAINER è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti occidentali (California settentrionale)
 - US West (Oregon)
 - Asia Pacifico (Hong Kong)
 - Asia Pacifico (Giacarta)
 - Asia Pacific (Hyderabad)
 - Asia Pacifico (Mumbai)
 - Asia Pacific (Osaka)
 - Asia Pacific (Seul)
 - Asia Pacifico (Singapore)
 - Asia Pacifico (Sydney)
 - Asia Pacifico (Tokyo)
 - Canada (Centrale)
 - Cina (Pechino)
 - China (Ningxia)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Europa (Londra)
 - Europa (Milano)
 - Europa (Parigi)
- Europa (Spagna)

- Europa (Stoccolma)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- Il tipo di ambiente `WINDOWS_SERVER_2022_CONTAINER` è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - US West (Oregon)
 - Asia Pacifico (Sydney)
 - Asia Pacifico (Tokyo)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Sud America (San Paolo)
- Il tipo di ambiente `LINUX_EC2` (`BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti occidentali (California settentrionale)
 - Stati Uniti occidentali (Oregon)
 - Africa (Città del Capo)
 - Asia Pacifico (Hong Kong)
 - Asia Pacifico (Giacarta)
 - Asia Pacifico (Melbourne)
 - Europa (Zurigo)
 - Asia Pacific (Hyderabad)
 - Asia Pacifico (Mumbai)
 - Asia Pacific (Osaka)
 - Asia Pacific (Seul)
 - Asia Pacifico (Singapore)

- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Europa (Stoccolma)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti occidentali)
- AWS GovCloud (Stati Uniti orientali)
- Il tipo di ambiente ARM_EC2 (BUILD_GENERAL1_SMALL,BUILD_GENERAL1_MEDIUM,BUILD_GENERAL1_LARGE) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti occidentali (California settentrionale)
 - US West (Oregon)
 - Asia Pacifico (Hong Kong)
 - Asia Pacifico (Giacarta)
 - Europa (Zurigo)
 - Asia Pacific (Hyderabad)
 - Asia Pacifico (Mumbai)
 - Asia Pacific (Osaka)

- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Europa (Stoccolma)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti occidentali)
- AWS GovCloud (Stati Uniti orientali)
- Il tipo di ambiente `WINDOWS_EC2` (`BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Ohio)
 - Stati Uniti orientali (Virginia settentrionale)
 - US West (Oregon)
 - Asia Pacifico (Sydney)
 - Asia Pacifico (Tokyo)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Sud America (San Paolo)

- La modalità di elaborazione AWS Lambda (ARM_LAMBDA_CONTAINEReLINUX_LAMBDA_CONTAINER) è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti orientali (Ohio)
 - US West (Oregon)
 - Asia Pacifico (Mumbai)
 - Asia Pacifico (Singapore)
 - Asia Pacifico (Sydney)
 - Asia Pacifico (Tokyo)
 - Europa (Francoforte)
 - Europa (Irlanda)
 - Sud America (San Paolo)
- La modalità di elaborazione MAC_ARM è disponibile solo nelle seguenti regioni:
 - Stati Uniti orientali (Virginia settentrionale)
 - Stati Uniti orientali (Ohio)
 - US West (Oregon)
 - Asia Pacifico (Sydney)
 - Europa (Francoforte)

Per il tipo di calcolo BUILD_GENERAL1_2XLARGE, sono supportate le immagini Docker fino a 100 GB non compresse.

Note

Per immagini di ambiente di compilazione personalizzate, CodeBuild supporta immagini Docker fino a 50 GB non compresse in Linux e Windows, indipendentemente dal tipo di elaborazione. Per controllare la dimensione dell'immagine di compilazione, utilizzare Docker per eseguire il comando `docker images` *REPOSITORY:TAG*.

Puoi usare Amazon EFS per accedere a più spazio nel tuo contenitore di build. Per ulteriori informazioni, consulta [Esempio di Amazon Elastic File System per AWS CodeBuild](#). Se si desidera

manipolare lo spazio su disco del container durante la compilazione, la compilazione deve essere eseguita in modalità privilegiata.

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

Shell e comandi negli ambienti di compilazione

Fornisci un set di comandi AWS CodeBuild da eseguire in un ambiente di compilazione durante il ciclo di vita di una build (ad esempio, l'installazione delle dipendenze della build e il test e la compilazione del codice sorgente). Esistono vari modi per specificare tali comandi:

- Crea un file di specifiche di compilazione e includilo con il codice sorgente. In questo file, specifica i comandi che desideri eseguire in ciascuna fase del ciclo di vita della compilazione. Per ulteriori informazioni, consulta la [Riferimento alle specifiche di costruzione per CodeBuild](#).
- Usa la CodeBuild console per creare un progetto di compilazione. In Insert build commands (Inserisci comandi di compilazione), per Build commands (Comandi di compilazione), inserisci i comandi che desideri eseguire nella fase build. Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#).
- Usa la CodeBuild console per modificare le impostazioni di un progetto di compilazione. In Insert build commands (Inserisci comandi di compilazione), per Build commands (Comandi di compilazione), inserisci i comandi che desideri eseguire nella fase build. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).
- Usa AWS CLI o AWS SDKs per creare un progetto di compilazione o modificare le impostazioni di un progetto di compilazione. Fai riferimento al codice sorgente che contiene un file di specifiche di compilazione con i tuoi comandi, oppure specifica una singola stringa che include i contenuti di file di specifiche di compilazione equivalente. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione](#) o [Modifica le impostazioni del progetto di compilazione](#).
- Usa AWS CLI o AWS SDKs per avviare una build, specificando un file buildspec o una singola stringa che include il contenuto di un file buildspec equivalente. Per ulteriori informazioni, consulta la descrizione del valore buildspecOverride in [Esegui le build manualmente](#).

È possibile specificare qualsiasi comando di linguaggio di comando Shell (sh). Nella versione 0.1 di buildspec, CodeBuild esegue ogni comando Shell in un'istanza separata nell'ambiente di compilazione. Questo significa che ogni comando viene eseguito separatamente da tutti gli altri comandi. Pertanto, come impostazione predefinita, non è possibile eseguire un singolo comando che si basa sullo stato di qualsiasi comando precedente (per esempio, modificare le directory o impostare le variabili di ambiente). Per aggirare questo limite, suggeriamo di utilizzare la versione 0.2, che risolve questo problema. Se è necessario utilizzare la versione 0.1, suggeriamo gli approcci seguenti:

- Includi uno script shell nel codice sorgente che contiene i comandi che desideri eseguire in una singola istanza della shell predefinita. Per esempio, puoi includere un file denominato `my-script.sh` nel codice sorgente che contiene i comandi, come `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd;`. Quindi, nel file di specifiche di compilazione, specifica il comando `./my-script.sh`.
- Nel file di specifiche di compilazione o nell'impostazione Build commands (Comandi di compilazione) solo per la fase build, inserisci un singolo comando che include tutti i comandi che desideri eseguire in una singola istanza della shell predefinita (ad esempio, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

Se si verifica un CodeBuild errore, l'errore potrebbe essere più difficile da risolvere rispetto all'esecuzione di un singolo comando nella propria istanza della shell predefinita.

I comandi eseguiti in un'immagine Windows Server Core utilizzano la shell. PowerShell

Variabili di ambiente degli ambienti di compilazione

AWS CodeBuild fornisce diverse variabili di ambiente che puoi usare nei comandi di compilazione:

AWS_DEFAULT_REGION

La AWS regione in cui è in esecuzione la build (ad esempio, `us-east-1`). Questa variabile di ambiente è utilizzata principalmente dalla AWS CLI.

AWS_REGION

La AWS regione in cui è in esecuzione la build (ad esempio, `us-east-1`). Questa variabile di ambiente è utilizzata principalmente dalla AWS SDKs.

CODEBUILD_BATCH_BUILD_IDENTIFIER

L'identificatore della build in una build batch. Questo è specificato nel batch buildspec. Per ulteriori informazioni, consulta [the section called “Riferimento alle specifiche di compilazione in Batch”](#).

CODEBUILD_BUILD_ARN

L'Amazon Resource Name (ARN) della build (ad esempio, `arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

L' CodeBuild ID della build (ad esempio, `codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

L'identificatore dell'immagine di CodeBuild compilazione (ad esempio, `aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

Il numero di build corrente del progetto.

CODEBUILD_BUILD_SUCCEEDING

Se la build corrente ha esito positivo. Impostare 0 se la compilazione non sta riuscendo, oppure 1 se la compilazione sta riuscendo.

CODEBUILD_INITIATOR

L'entità che ha avviato la compilazione. Se è stata CodePipeline avviata la build, questo è il nome della pipeline (ad esempio, `codepipeline/my-demo-pipeline`). Se un utente ha avviato la build, questo è il nome dell'utente (ad esempio, `MyUserName`). Se il plugin Jenkins for CodeBuild ha avviato la build, questa è la stringa `CodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

L'identificatore della AWS KMS chiave utilizzata per crittografare l'elemento di output CodeBuild della build (ad esempio o). `arn:aws:kms:region-ID:account-ID:key/key-ID alias/key-alias`

CODEBUILD_PROJECT_ARN

L'Amazon Resource Name (ARN) del progetto (ad esempio, `arn:aws:codebuild:region-ID:account-ID:project/project-name`).

CODEBUILD_PUBLIC_BUILD_URL

L'URL dei risultati della build per questa build sul sito web Public Builds. Questa variabile è impostata solo se il progetto di compilazione ha le build pubbliche abilitate. Per ulteriori informazioni, consulta [Get public build project URLs](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

L'identificatore di versione del codice sorgente di una build. Il contenuto dipende dal repository del codice sorgente:

CodeCommit GitHub, GitHub Enterprise Server e Bitbucket

Questa variabile contiene l'ID di commit.

CodePipeline

Questa variabile contiene la revisione del codice sorgente fornita da CodePipeline.

Se non CodePipeline è in grado di risolvere la revisione del codice sorgente, ad esempio quando l'origine è un bucket Amazon S3 che non ha il controllo delle versioni abilitato, questa variabile di ambiente non è impostata.

Simple Storage Service (Amazon S3)

Questa variabile non è impostata.

Se applicabile, la CODEBUILD_RESOLVED_SOURCE_VERSION variabile è disponibile solo dopo la DOWNLOAD_SOURCE fase.

CODEBUILD_SOURCE_REPO_URL

L'URL dell'artefatto di input o del repository del codice sorgente. Per Amazon S3, questo è s3:// seguito dal nome del bucket e dal percorso dell'elemento di input. Per CodeCommit e GitHub, questo è l'URL clone del repository. Se una build ha origine da CodePipeline, questa variabile di ambiente potrebbe essere vuota.

Per le fonti secondarie, la variabile di ambiente per l'URL del repository di origine secondario è CODEBUILD_SOURCE_REPO_URL_<sourceIdentifier>, dove si <sourceIdentifier> trova l'identificatore di origine creato.

CODEBUILD_SOURCE_VERSION

Il formato del valore dipende dal repository di origine.

- Per Amazon S3, è l'ID di versione associato all'elemento di input.
- Infatti CodeCommit, è l'ID di commit o il nome del ramo associato alla versione del codice sorgente da creare.
- Per GitHub GitHub Enterprise Server e Bitbucket è l'ID di commit, il nome del ramo o il nome del tag associato alla versione del codice sorgente da creare.

**Note**

Per una build GitHub di GitHub Enterprise Server attivata da un evento di pull request di webhook, lo è. `pr/pull-request-number`

Per le fonti secondarie, la variabile di ambiente per la versione di origine secondaria è `CODEBUILD_SOURCE_VERSION_<sourceIdentifier>`, `<sourceIdentifier>` dov'è l'identificatore di origine creato. Per ulteriori informazioni, consulta [Diverse origini di input ed esempi di artefatti di output](#).

CODEBUILD_SRC_DIR

Il percorso della directory CodeBuild utilizzato per la compilazione (ad esempio,). `/tmp/src123456789/src`

Per le fonti secondarie, la variabile di ambiente per il percorso della directory di origine secondaria è `CODEBUILD_SRC_DIR_<sourceIdentifier>`, dove si `<sourceIdentifier>` trova l'identificatore di origine che crei. Per ulteriori informazioni, consulta [Diverse origini di input ed esempi di artefatti di output](#).

CODEBUILD_START_TIME

L'ora di inizio della build specificata come timestamp Unix in millisecondi.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

L'ID dell'account dell'utente che ha attivato l'evento webhook.

CODEBUILD_WEBHOOK_BASE_REF

Il nome di riferimento di base dell'evento webhook che attiva la build corrente. Per una richiesta pull, questo è il riferimento del ramo.

CODEBUILD_WEBHOOK_EVENT

L'evento webhook che attiva la build corrente.

CODEBUILD_WEBHOOK_MERGE_COMMIT

L'identificatore del merge commit utilizzato per la build. Questa variabile viene impostata quando una pull request di Bitbucket viene unita alla strategia squash e il ramo pull request viene chiuso. In questo caso, il commit di pull request originale non esiste più, quindi questa variabile di ambiente contiene l'identificatore dello squashed merge commit.

CODEBUILD_WEBHOOK_PREV_COMMIT

L'ID del commit più recente prima dell'evento webhook push che attiva la build corrente.

CODEBUILD_WEBHOOK_HEAD_REF

Il nome di riferimento principale dell'evento webhook che attiva la build corrente. Si tratta di un riferimento ramo o tag.

CODEBUILD_WEBHOOK_TRIGGER

Mostra l'evento webhook che ha attivato la build. Questa variabile è disponibile solo per build attivate da un webhook. Il valore viene analizzato dal payload inviato a CodeBuild by GitHub, GitHub Enterprise Server o Bitbucket. Il formato del valore dipende dal tipo di evento che ha attivato la build.

- Per le build attivate da una richiesta di pull, è `pr/pull-request-number`.
- Per le build attivate dalla creazione di un nuovo ramo o dal push di un commit a un ramo, è `branch/branch-name`.
- Per le build attivate dal push di un tag a un repository, è `tag/tag-name`.

CASA

Questa variabile di ambiente è sempre impostata su `/root`.

AWS CodeBuild supporta anche una serie di variabili di ambiente per build di runner ospitate autonomamente. Per ulteriori informazioni su CodeBuild self-hosted runner, consulta [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#)

CODEBUILD_RUNNER_OWNER

Il proprietario del repository che attiva la build del runner ospitato autonomamente.

CODEBUILD_RUNNER_REPO

Il nome del repository che attiva la build del runner ospitato autonomamente.

CODEBUILD_RUNNER_REPO_DOMAIN

Il dominio del repository che attiva la build del runner ospitato autonomamente. Solo build Enterprise specificate. GitHub

CODEBUILD_WEBHOOK_LABEL

L'etichetta utilizzata per configurare le sostituzioni di build e il runner self-hosted durante la compilazione.

CODEBUILD_WEBHOOK_RUN_ID

L'ID di esecuzione del flusso di lavoro associato alla build.

CODEBUILD_WEBHOOK_JOB_ID

L'ID del lavoro associato alla build.

CODEBUILD_WEBHOOK_WORKFLOW_NAME

Il nome del flusso di lavoro associato alla build, se presente nel payload della richiesta del webhook.

CODEBUILD_RUNNER_WITH_BUILDSPEC

Se un override buildspec è configurato nelle etichette di richiesta runner self-hosted, questo è impostato su. `true`

È anche possibile fornire ambienti di compilazione con le proprie variabili di ambiente. Per ulteriori informazioni, consulta i seguenti argomenti:

- [Usa CodeBuild con CodePipeline](#)
- [Creazione di un progetto di compilazione](#)
- [Modifica le impostazioni del progetto di compilazione](#)
- [Esegui le build manualmente](#)
- [Riferimento per buildspec](#)

Per elencare tutte le variabili di ambiente disponibili in un ambiente di compilazione, è possibile eseguire il comando `printenv` (per un ambiente di compilazione basato su Linux) o `"Get-ChildItem Env:"` (per ambienti di compilazione basati su Windows) durante una compilazione.

Ad eccezione di quelle elencate in precedenza, le variabili di ambiente che iniziano con sono per uso interno. CODEBUILD_ CodeBuild Non devono essere utilizzate nei comandi di compilazione.

Important

Sconsigliamo vivamente l'uso di variabili di ambiente per memorizzare valori sensibili, in particolare la chiave IDs di AWS accesso. Le variabili di ambiente possono essere visualizzate in testo semplice utilizzando strumenti come la CodeBuild console e il AWS CLI. Ti consigliamo di archiviare i valori sensibili in Amazon EC2 Systems Manager Parameter Store e poi recuperarli dalle tue specifiche di build. Per memorizzare valori sensibili, consulta [Systems Manager Parameter Store](#) and [Walkthrough: Create and test a String parameter \(console\)](#) nella Amazon EC2 Systems Manager User Guide. Per recuperarli, consultare la mappatura parameter-store in [Sintassi buildspec](#).

CODEBUILD_BUILD_URL

L'URL dei risultati della build per questa build.

Attività in background degli ambienti di compilazione

È possibile eseguire attività in background negli ambienti di compilazione. Per fare ciò, nelle specifiche di compilazione, utilizzare il comando `nohup` per eseguire un comando come attività in background, anche se il processo di compilazione esce dalla shell. Utilizza il comando `disown` per forzare l'arresto dell'esecuzione di un'attività in background.

Esempi:

- Avviare un processo in background e attenderne il completamento in un secondo momento:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Avviare un processo in background e non attenderne mai il completamento:

```
|  
nohup sleep 30 & disown $!
```

- Avviare un processo in background e terminarlo in un secondo momento:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
kill $(cat pidfile)
```

Costruisci progetti

Un progetto di compilazione include informazioni su come eseguire una build, tra cui dove trovare il codice sorgente, quale ambiente di compilazione utilizzare, quali comandi di build eseguire e dove archiviare l'output della build.

Puoi eseguire queste attività durante l'utilizzo di progetti di compilazione:

Argomenti

- [Crea un progetto di compilazione in AWS CodeBuild](#)
- [Creazione di una regola di notifica](#)
- [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#)
- [Token di accesso multiplo in CodeBuild](#)
- [Eliminare progetti di compilazione in AWS CodeBuild](#)
- [Get public build project URLs](#)
- [Condividi progetti di costruzione](#)
- [Etichetta: costruisci progetti](#)
- [Usa i runner con AWS CodeBuild](#)
- [Usa i webhook con AWS CodeBuild](#)
- [Visualizza i dettagli di un progetto di costruzione in AWS CodeBuild](#)
- [Visualizza i nomi dei progetti di compilazione in AWS CodeBuild](#)

Crea un progetto di compilazione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs creare un progetto di compilazione.
AWS CLI

Argomenti

- [Prerequisiti](#)
- [Creare un progetto di compilazione \(console\)](#)
- [Creazione di un progetto di compilazione \(AWS CLI\)](#)
- [Creazione di un progetto di compilazione \(AWS SDKs\)](#)
- [Creazione di un progetto di compilazione \(CloudFormation\)](#)

Prerequisiti

Prima di creare un progetto di compilazione, rispondi alle domande contenute in [Pianifica una compilazione](#).

Creare un progetto di compilazione (console)

Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.

Scegliere Create build project (Crea progetto di compilazione).

Compila le seguenti sezioni. Una volta completato, scegli Crea progetto di compilazione nella parte inferiore della pagina.

Sezioni:

- [Configurazione del progetto](#)
- [Origine](#)
- [Ambiente](#)
- [Buildspec](#)
- [Configurazione Batch](#)
- [Artifacts](#)
- [Log](#)

Configurazione del progetto

Project name (Nome progetto)

Inserisci un nome per questo progetto di compilazione. I nomi dei progetti di build devono essere univoci per ogni AWS account.

Descrizione

Inserisci una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.

Costruisci un badge

(Facoltativo) Seleziona Abilita il badge di costruzione per rendere visibile e incorporabile lo stato di costruzione del progetto. Per ulteriori informazioni, consulta [Creazione di esempi di badge](#).

Note

Il badge Build non si applica se il tuo provider di origine è Amazon S3.

Abilita il limite di compilazione simultanea

(Facoltativo) Se desideri limitare il numero di build simultanee per questo progetto, esegui i seguenti passaggi:

1. Seleziona Limita il numero di build simultanee che questo progetto può avviare.
2. In Limite di build simultanee, inserisci il numero massimo di build simultanee consentite per questo progetto. Questo limite non può essere superiore al limite di creazione simultanea impostato per l'account. Se si tenta di inserire un numero superiore al limite dell'account, viene visualizzato un messaggio di errore.

Le nuove compilazione vengono avviate solo se il numero corrente di compilazioni è minore o uguale a questo limite. Se il conteggio delle compilazioni correnti soddisfa questo limite, le nuove compilazioni vengono limitate e non vengono eseguite.

Informazioni aggiuntive

(Facoltativo) Per i tag, inserisci il nome e il valore di tutti i tag che desideri vengano utilizzati AWS dai servizi di supporto. Utilizzare Add row (Aggiungi riga) per aggiungere un tag. Puoi aggiungere fino a 50 tag.

Origine

Fornitore di origine

Scegli il tipo di provider del codice sorgente. Utilizza i seguenti elenchi per effettuare le selezioni appropriate per il tuo provider di origine:

 Note

CodeBuild non supporta Bitbucket Server.

Amazon S3

Bucket

Scegli il nome del bucket di input che contiene il codice sorgente.

Chiave oggetto S3 o cartella S3

Inserisci il nome del file ZIP o il percorso della cartella che contiene il codice sorgente. Inserisci una barra (/) per scaricare tutti i contenuti del bucket S3.

Versione sorgente

Immettete l'ID di versione dell'oggetto che rappresenta la build del file di input. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

CodeCommit

Repository

Scegli il repository che desideri utilizzare.

Tipo di riferimento

Scegli Branch, Git tag o Commit ID per specificare la versione del codice sorgente. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

 Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Scegli di creare un clone poco profondo con una cronologia troncata al numero di commit specificato. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Bitbucket

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli CodeConnections la OAuthpassword dell'app o il token di accesso personale a cui connetterti CodeBuild.

Connessione

Seleziona una connessione Bitbucket o un segreto Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio account Bitbucket o Archivio pubblico e inserisci l'URL del repository.

Versione sorgente

Inserisci un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#)

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona **Use Git submodules** (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona **Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build** se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto **Status**, inserisci il valore da utilizzare per il `name` parametro nello stato del commit di Bitbucket. Per ulteriori informazioni, consulta [compilazione](#) nella documentazione di riferimento dell'API Bitbucket.

Per **Target URL**, inserisci il valore da utilizzare per il `url` parametro nello stato del commit di Bitbucket. Per ulteriori informazioni, consulta [compilazione](#) nella documentazione di riferimento dell'API Bitbucket.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

Negli eventi del webhook di origine primaria, seleziona **Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository** se desideri creare il codice sorgente ogni volta che viene inviata una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta [Eventi webhook Bitbucket](#)

GitHub

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli GitHub App o Token di accesso personale a cui connetterti CodeBuild. OAuth

Connessione

Seleziona una GitHub connessione o un segreto di Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio GitHub account, Public repository o GitHub scoped webhook e inserisci l'URL del repository.

Versione sorgente

Inserisci un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#)

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto Status, immettete il valore da utilizzare per il `context` parametro nel GitHub commit status. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Per Target URL, inserisci il valore da utilizzare per il `target_url` parametro nello stato del GitHub commit. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

Negli eventi del webhook di origine primaria, seleziona Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository se desideri creare il codice sorgente ogni volta che viene inviata una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta [GitHub eventi webhook](#)

GitHub Enterprise Server

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli il CodeConnectionsnostro token di accesso personale a cui connetterti CodeBuild.

Connessione

Seleziona una connessione GitHub Enterprise o un segreto Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio account GitHub Enterprise o GitHub Enterprise scoped webhook e inserisci l'URL del repository.

Versione sorgente

Inserisci una pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto Status, immettete il valore da utilizzare per il `context` parametro nel GitHub commit status. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Per Target URL, inserisci il valore da utilizzare per il `target_url` parametro nello stato del GitHub commit. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

SSL non sicuro

Seleziona Abilita SSL non sicuro per ignorare gli avvisi SSL durante la connessione al repository del tuo progetto Enterprise. GitHub

In Primary source webhook events, seleziona Ricostruisci ogni volta che viene inserita una modifica al codice in questo repository se desideri creare il codice sorgente ogni volta che viene inserita una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta. [GitHub eventi webhook](#)

GitLab

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

CodeConnections viene utilizzato per connettersi GitLab a CodeBuild.

Connessione

Seleziona una GitLab connessione tramite la quale connetterti CodeConnections.

Repository

Scegli il repository che desideri utilizzare.

Versione sorgente

Inserisci un ID di pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Stato della build

Seleziona Segnala lo stato della compilazione al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

GitLab Self Managed

Credenziale

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

CodeConnectionsviene utilizzato per connettere GitLab Self Managed a CodeBuild.

Connessione

Seleziona una connessione GitLab Self Managed tramite la quale connetterti CodeConnections.

Repository

Scegli il repository che desideri utilizzare.

Versione sorgente

Inserisci un ID di pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Stato della build

Seleziona Segnala lo stato della compilazione al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Ambiente

Modello di provisioning

Esegui una di queste operazioni:

- Per utilizzare flotte on-demand gestite da AWS CodeBuild, scegli On-demand. Con le flotte on-demand, CodeBuild fornisce elaborazione per le tue build. Le macchine vengono distrutte al termine della costruzione. Le flotte on-demand sono completamente gestite e includono funzionalità di scalabilità automatica per gestire i picchi di domanda.
- Per utilizzare flotte a capacità riservata gestite da AWS CodeBuild, scegli Capacità riservata, quindi seleziona il nome di una flotta. Con le flotte a capacità riservata, puoi configurare un set di istanze dedicate per il tuo ambiente di costruzione. Queste macchine rimangono inattive, pronte per l'elaborazione immediata di build o test e riducono la durata di costruzione. Con flotte a capacità riservata, le macchine sono sempre in funzione e continueranno a sostenere costi finché verranno rifornite.

Per informazioni, consultare [Run si basa su flotte a capacità riservata](#).

Immagine dell'ambiente

Esegui una di queste operazioni:

- Per utilizzare un'immagine Docker gestita da AWS CodeBuild, scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Immagine e Versione dell'immagine. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).
- Per utilizzare un'altra immagine Docker, selezionare Custom image (Immagine personalizzata). Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS
- Per utilizzare un'immagine Docker privata, scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

Note

CodeBuild sostituisce le immagini Docker ENTRYPOINT personalizzate.

Calcolo

Esegui una di queste operazioni:

- Per usare EC2 compute, scegli. EC2 EC2 compute offre una flessibilità ottimizzata durante le operazioni.
- Per usare l'elaborazione Lambda, scegli Lambda. L'elaborazione Lambda offre velocità di avvio ottimizzate per le tue build. Lambda supporta build più veloci grazie a una minore latenza di avvio. Lambda inoltre si ridimensiona automaticamente, quindi le build non devono attendere in coda per essere eseguite. Per informazioni, consultare [Esegui build su calcolo AWS Lambda](#).

Ruolo di servizio

Esegui una di queste operazioni:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.
- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

Note

Quando usi la console per creare un progetto di build, puoi creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

Configurazione aggiuntiva

Limite di tentativi automatici

Specificate il numero di tentativi automatici aggiuntivi dopo una compilazione non riuscita. Ad esempio, se il limite di tentativi automatici è impostato su 2, CodeBuild chiamerà l'RetryBuildAPI per riprovare automaticamente la build per un massimo di 2 volte aggiuntive.

Timeout

Specificate un valore, compreso tra 5 minuti e 36 ore, dopo il quale CodeBuild interrompe la compilazione se non è completa. Se `hours` (ore) e `minutes` (minuti) vengono lasciati vuoti, viene utilizzato il valore predefinito di 60 minuti.

Privilegiato

(Facoltativo) Seleziona Abilita questo flag se desideri creare immagini Docker o desideri che le tue build ottengano privilegi elevati solo se prevedi di utilizzare questo progetto di compilazione per creare immagini Docker. In caso contrario, tutte le compilazioni associate che tentano di interagire con il daemon Docker avranno esito negativo. È inoltre necessario avviare il daemon Docker in modo che le compilazioni possano interagire con esso. Un modo per fare ciò è inizializzare il daemon Docker nella fase `install` delle specifiche di compilazione eseguendo i comandi di compilazione seguenti. Non eseguire questi comandi se hai scelto un'immagine dell'ambiente di compilazione fornita dal supporto Docker. CodeBuild

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Se vuoi CodeBuild lavorare con il tuo VPC:

- Per VPC, scegli l'ID VPC che utilizza. CodeBuild
- Per le sottoreti VPC, scegli le sottoreti che includono le risorse che utilizza. CodeBuild
- Per i gruppi di sicurezza VPC, scegli i gruppi di sicurezza da CodeBuild utilizzare per consentire l'accesso alle risorse in VPCs

Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Calcolo

Scegli una delle opzioni disponibili.

Credenziali di registro

Specificare una credenziale di registro quando il progetto è configurato con un'immagine di registro non privata.

Note

Questa credenziale verrà utilizzata solo se le immagini vengono sostituite con quelle dei registri privati.

Variabili di ambiente

Immettete il nome e il valore, quindi scegliete il tipo di ogni variabile di ambiente da utilizzare per le build.

Note

CodeBuild imposta automaticamente la variabile di ambiente per la tua AWS regione. È necessario impostare le seguenti variabili di ambiente se non le hai aggiunte al tuo buildspec.yml:

- AWS_ACCOUNT_ID
- IMAGE_REPO_NAME
- IMAGE_TAG

La console e AWS CLI gli utenti possono visualizzare le variabili di ambiente. Se la visibilità della variabile di ambiente non rappresenta un problema, configurare i campi Name (Nome) e Value (Valore), quindi impostare Type (Tipo) su Plaintext (Testo normale).

Ti consigliamo di memorizzare una variabile di ambiente con un valore sensibile, come un ID di chiave di AWS accesso, una chiave di accesso AWS segreta o una password come parametro in Amazon EC2 Systems Manager Parameter Store o Gestione dei segreti AWS.

Se utilizzi Amazon EC2 Systems Manager Parameter Store, per Tipo scegli Parameter. Per Nome, inserisci un identificativo CodeBuild a cui fare riferimento. Per Value, inserisci il nome

del parametro memorizzato in Amazon EC2 Systems Manager Parameter Store. Utilizzando un parametro esemplificativo denominato `/CodeBuild/dockerLoginPassword`, per Type (Tipo) selezionare Parameter (Parametro). Per Nome, immetti `LOGIN_PASSWORD`. In Valore, specifica `/CodeBuild/dockerLoginPassword`.

Important

Se utilizzi Amazon EC2 Systems Manager Parameter Store, ti consigliamo di memorizzare i parametri con nomi di parametri che iniziano con `/CodeBuild/` (ad esempio, `/CodeBuild/dockerLoginPassword`). Puoi utilizzare la CodeBuild console per creare un parametro in Amazon EC2 Systems Manager. Selezionare Create a parameter (Crea un parametro), quindi seguire le istruzioni nella finestra di dialogo. (In quella finestra di dialogo, per la chiave KMS, puoi specificare l'ARN di AWS KMS una chiave nel tuo account. Amazon EC2 Systems Manager utilizza questa chiave per crittografare il valore del parametro durante l'archiviazione e decrittografarlo durante il recupero.) Se usi la CodeBuild console per creare un parametro, la console avvia il nome del parametro con il `/CodeBuild/` momento in cui viene memorizzato. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.

Se il progetto di compilazione si riferisce a parametri memorizzati in Amazon EC2 Systems Manager Parameter Store, il ruolo di servizio del progetto di compilazione deve consentire l'azione `ssm:GetParameters`. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se invece hai scelto Existing service role (Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build si riferisce a parametri memorizzati in Amazon EC2 Systems Manager Parameter Store con nomi di parametri che non iniziano con `/CodeBuild/` e hai scelto Nuovo ruolo di servizio, devi aggiornare quel ruolo di servizio per consentire l'accesso ai nomi dei parametri che non iniziano con `/CodeBuild/`. Ciò dipende dal fatto che tale ruolo consente l'accesso solo a nomi di parametro che iniziano con `/CodeBuild/`.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i parametri nello `/CodeBuild/` spazio dei nomi in Amazon EC2 Systems Manager Parameter Store.

Le variabili di ambiente impostate sostituiscono quelle esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata MY_VAR con un valore di my_value, e si imposta una variabile di ambiente denominata MY_VAR con un valore di other_value, il valore my_value viene sostituito da other_value. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata PATH con un valore di /usr/local/sbin:/usr/local/bin, e si imposta una variabile di ambiente denominata PATH con un valore di \$PATH:/usr/share/ant/bin, il valore di /usr/local/sbin:/usr/local/bin viene sostituito dal valore letterale \$PATH:/usr/share/ant/bin.

Non impostare variabili di ambiente con nomi che iniziano con CODEBUILD_. Questo prefisso è riservato per l'uso interno .

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità.
- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità.
- Il valore della dichiarazione buildspec ha la minima priorità.

Se usi Secrets Manager, per Tipo, scegli Secrets Manager. Per Nome, inserite un identificatore CodeBuild a cui fare riferimento. Per Valore, immettere un reference-key utilizzando il modello `secret-id:json-key:version-stage:version-id`. Per informazioni, consultare [Secrets Manager reference-key in the buildspec file](#).

Important

Se utilizzi Secrets Manager, ti consigliamo di archiviare i segreti con nomi che iniziano con /CodeBuild/ (ad esempio, /CodeBuild/dockerLoginPassword). Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager, il ruolo di servizio del progetto di build deve consentire l'azione `secretsmanager:GetSecretValue`. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se invece hai scelto Existing service role

(Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager con nomi segreti che non iniziano con `/CodeBuild/` e hai scelto Nuovo ruolo di servizio, devi aggiornare il ruolo di servizio per consentire l'accesso ai nomi segreti che non iniziano con `/CodeBuild/`. Questo perché il ruolo di servizio consente l'accesso solo ai nomi segreti che iniziano con `/CodeBuild/`.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i segreti nello spazio dei `/CodeBuild/` nomi in Secrets Manager.

Buildspec

Specifiche di costruzione

Esegui una di queste operazioni:

- Se il codice sorgente include un file di specifiche di compilazione, selezionare Use a buildspec file (Utilizza un file buildspec). Per impostazione predefinita, CodeBuild cerca un file denominato `buildspec.yml` nella directory principale del codice sorgente. Se il tuo file buildspec utilizza un nome o una posizione diversi, inserisci il percorso dalla radice di origine in Buildspec name (ad esempio, `o. buildspec-two.yml configuration/buildspec.yml`). Se il file buildspec si trova in un bucket S3, deve trovarsi nella stessa regione del progetto di build. AWS Specificate il file buildspec utilizzando il relativo ARN (ad esempio, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Se il codice sorgente non include un file di specifiche di compilazione o se desideri eseguire comandi di compilazione diversi da quelli specificati per la fase build del file `buildspec.yml` nella directory radice del codice sorgente, scegli Insert build commands (Inserisci comandi di compilazione). Per Build commands (Comandi di compilazione), immetti i comandi che desideri eseguire nella fase build. In caso di più comandi, separa ogni comando con `&&` (ad esempio `mvn test && mvn package`). Per eseguire comandi in altre fasi, o se disponi di un lungo elenco di comandi per la build fase, aggiungi un `buildspec.yml` file alla directory principale del codice sorgente, aggiungi i comandi al file e quindi scegli Usa il buildspec.yml nella directory principale del codice sorgente.

Per ulteriori informazioni, consulta la [Riferimento per buildspec](#).

Configurazione Batch

È possibile eseguire un gruppo di build come singola operazione. Per ulteriori informazioni, consulta [Esegui le build in batch](#).

Definire la configurazione in batch

Seleziona questa opzione per consentire le compilazioni in batch in questo progetto.

Ruolo del servizio Batch

Fornisce il ruolo di servizio per le build in batch.

Seleziona una delle seguenti opzioni:

- Se non disponi di un ruolo di servizio batch, scegli Nuovo ruolo di servizio. In Ruolo di servizio, inserisci un nome per il nuovo ruolo.
- Se disponi di un ruolo di servizio batch, scegli Ruolo di servizio esistente. In Ruolo di servizio, scegli il ruolo di servizio.

Le build batch introducono un nuovo ruolo di sicurezza nella configurazione batch. Questo nuovo ruolo è obbligatorio in quanto è CodeBuild necessario poter richiamare `StartBuildStopBuild`, e `RetryBuild` le azioni per conto dell'utente per eseguire le build come parte di un batch. I clienti devono utilizzare un nuovo ruolo e non lo stesso ruolo utilizzato nella build, per due motivi:

- L'assegnazione del ruolo `StartBuild` di build e `RetryBuild` delle autorizzazioni consentirebbe a una singola build di avviare più build tramite `buildspec`. `StopBuild`
- CodeBuild le build in batch forniscono restrizioni che limitano il numero di build e i tipi di calcolo che possono essere utilizzati per le build del batch. Se il ruolo di build dispone di queste autorizzazioni, è possibile che le build stesse possano aggirare queste restrizioni.

Tipi di calcolo consentiti per batch

Seleziona i tipi di calcolo consentiti per il batch. Seleziona tutte le risposte pertinenti.

Flotte consentite per lotto

Seleziona le flotte consentite per il batch. Seleziona tutte le risposte pertinenti.

Numero massimo di build consentite in batch

Immettere il numero massimo di build consentite nel batch. Se un batch supera questo limite, il batch avrà esito negativo.

Timeout del batch

Immettere il periodo di tempo massimo per il completamento della creazione del batch.

Combina artefatti

Seleziona Combina tutti gli artefatti del batch in un'unica posizione per combinare tutti gli artefatti del batch in un'unica posizione.

Modalità report Batch

Seleziona la modalità di report sullo stato della build desiderata per le compilazioni in batch.

Note

Questo campo è disponibile solo quando l'origine del progetto è Bitbucket o GitHub Enterprise e in Origine è selezionato Segnala lo stato della build al fornitore di origine quando le build iniziano e finiscono. GitHub

Building aggregate

Seleziona questa opzione per combinare gli stati di tutte le build del batch in un unico rapporto sullo stato.

Building individuali

Seleziona questa opzione per far sì che gli stati di compilazione di tutte le build del batch vengano riportati separatamente.

Artifacts

Tipo

Esegui una di queste operazioni:

- Se non si desidera creare alcun artefatto di output di compilazione, scegliere No artifacts (Nessun artefatto). Questa operazione potrebbe essere utile se esegui solo test di compilazione o desideri inviare un'immagine Docker a un repository Amazon ECR.
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Se

desideri eseguire l'output di un file ZIP e vuoi che il file ZIP abbia un'estensione di file, assicurati di includerla dopo il nome del file ZIP.

- Selezionare Enable semantic versioning (Abilita funzione Versioni multiple semantica) se si desidera che un nome specificato nel file buildspec sostituisca qualsiasi nome specificato nella console. Il nome del file di specifiche di compilazione viene calcolato in fase di compilazione e utilizza il linguaggio di comando Shell. Ad esempio, è possibile aggiungere una data e un'ora al nome dell'artefatto in modo che sia sempre univoco. I nomi di artefatto univoci impediscono che gli artefatti vengano sovrascritti. Per ulteriori informazioni, consulta [Sintassi buildspec](#).
- Per Bucket name (Nome bucket) selezionare il nome del bucket di output.
- Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, `appspec.yml`, `target/my-app.jar`). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).
- Se desideri crittografare gli artefatti di compilazione, seleziona Remove artifacts encryption (Rimuovi crittografia artefatti).

Per ogni insieme secondario di artefatti desiderato:

1. Per Artifact identifier (Identificatore di artefatto), inserire un valore inferiore a 128 caratteri e che contiene solo caratteri alfanumerici e trattini bassi.
2. Scegliere Add artifact (Aggiungi artefatto).
3. Seguire le fasi precedenti configurare gli artefatti secondari.
4. Selezionare Save artifact (Salva elemento).

Configurazione aggiuntiva

Chiave di crittografia

(Facoltativo) Scegliere una delle seguenti operazioni:

- Per utilizzare la chiave di crittografia Chiave gestita da AWS per Amazon S3 nel tuo account per crittografare gli artefatti di output della build, lascia vuota la chiave di crittografia. Questa è l'impostazione predefinita.

- Per utilizzare una chiave gestita dal cliente per crittografare gli artefatti di output della build, in Chiave di crittografia, inserisci l'ARN della chiave KMS. Utilizzare il formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo di cache

In Cache type (Tipo di cache), procedere in uno dei modi seguenti:

- Se non si desidera utilizzare una cache, scegliere No cache (Nessuna cache).
- Se desideri utilizzare una cache Amazon S3, scegli Amazon S3 e procedi come segue:
 - Per Bucket, selezionare il nome del bucket S3 in cui è archiviata la cache.
 - (Facoltativo) Per il prefisso del percorso della cache, inserisci un prefisso del percorso Amazon S3. Il valore Cache path prefix (Prefisso percorso cache) è simile a un nome di directory. Consente di archiviare la cache sotto la stessa directory in un bucket.

Important

Non aggiungere una barra finale (/) alla fine del prefisso del percorso.

- Se si desidera utilizzare una cache locale, selezionare Local (Locale), quindi scegliere una o più modalità cache locali.

Note

La modalità Cache di livello Docker è disponibile solo per Linux. Se la scegli, il progetto deve essere eseguito in modalità privilegiata.

L'utilizzo di una cache consente di risparmiare tempo di compilazione perché alcune parti riutilizzabili dell'ambiente di compilazione vengono memorizzate nella cache e utilizzate in compilazioni diverse. Per informazioni su come specificare una cache nel file di specifiche di compilazione, consulta [Sintassi buildspec](#). Per ulteriori informazioni sul caching, consulta [Creazioni di cache per migliorare le prestazioni](#).

Log

Scegli i log che desideri creare. Puoi creare Amazon CloudWatch Logs, Amazon S3 log o entrambi.

CloudWatch

Se desideri i log di Amazon CloudWatch Logs:

CloudWatch logs

Seleziona i logCloudWatch .

Group name (Nome gruppo)

Inserisci il nome del tuo gruppo di CloudWatch log Amazon Logs.

Nome dello stream

Inserisci il nome del tuo flusso di CloudWatch log di Amazon Logs.

S3

Se desideri i log di Amazon S3:

Registri S3

Selezionare S3 logs (Log S3).

Bucket

Scegli il nome del bucket S3 per i tuoi log.

Prefisso del percorso

Inserisci il prefisso per i tuoi log.

Disattiva la crittografia dei log S3

Seleziona se non desideri che i registri S3 siano crittografati.

Creazione di un progetto di compilazione (AWS CLI)

Per ulteriori informazioni sull'utilizzo di AWS CLI with CodeBuild, consulta. [Guida di riferimento alla riga di comando](#)

Per creare un progetto di CodeBuild compilazione utilizzando AWS CLI, crei una struttura di [progetto](#) in formato JSON, compili la struttura e chiami il [create-project](#) comando per creare il progetto.

Crea il file JSON

Crea un file JSON scheletrico con il [create-project](#) comando, utilizzando l'opzione: --generate-cli-skeleton

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Questo crea un file JSON con il percorso e il nome del file specificati da. *<json-file>*

Compila il file JSON

Modifica i dati JSON come segue e salva i risultati.

```
{
  "name": "<project-name>",
  "description": "<description>",
  "source": {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" | "GITLAB" |
    "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "buildStatusConfig": {
      "context": "<context>",
      "targetUrl": "<target-url>"
    },
    "gitSubmodulesConfig": {
      "fetchSubmodules": "<fetch-submodules>"
    },
    "auth": {
      "type": "<auth-type>",
      "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
      "GITLAB" | "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
      "location": "<source-location>",
      "gitCloneDepth": "<git-clone-depth>",
      "buildspec": "<buildspec>",
      "InsecureSsl": "<insecure-ssl>",
      "reportBuildStatus": "<report-build-status>",
      "auth": {
        "type": "<auth-type>",
```

```

        "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
}
],
"secondarySourceVersions": [
{
    "sourceIdentifier": "<secondary-source-identifier>",
    "sourceVersion": "<secondary-source-version>"
}
],
"sourceVersion": "<source-version>",
"artifacts": {
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<artifacts-location>",
    "path": "<artifacts-path>",
    "namespaceType": "<artifacts-namespaceType>",
    "name": "<artifacts-name>",
    "overrideArtifactName": "<override-artifact-name>",
    "packaging": "<artifacts-packaging>"
},
"secondaryArtifacts": [
{
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<secondary-artifact-location>",
    "path": "<secondary-artifact-path>",
    "namespaceType": "<secondary-artifact-namespaceType>",
    "name": "<secondary-artifact-name>",
    "packaging": "<secondary-artifact-packaging>",
    "artifactIdentifier": "<secondary-artifact-identifier>"
}
],
"cache": {
    "type": "<cache-type>",
    "location": "<cache-location>",
    "mode": [
        "<cache-mode>"
    ]
},
"environment": {
    "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
"WINDOWS_SERVER_2019_CONTAINER" | "WINDOWS_SERVER_2022_CONTAINER",
    "image": "<image>",

```

```

    "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
    "BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
    "certificate": "<certificate>",
    "environmentVariables": [
      {
        "name": "<environmentVariable-name>",
        "value": "<environmentVariable-value>",
        "type": "<environmentVariable-type>"
      }
    ],
    "registryCredential": [
      {
        "credential": "<credential-arn-or-name>",
        "credentialProvider": "<credential-provider>"
      }
    ],
    "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
    "privilegedMode": "<privileged-mode>"
  },
  "serviceRole": "<service-role>",
  "autoRetryLimit": <auto-retry-limit>,
  "timeoutInMinutes": <timeout>,
  "queuedTimeoutInMinutes": <queued-timeout>,
  "encryptionKey": "<encryption-key>",
  "tags": [
    {
      "key": "<tag-key>",
      "value": "<tag-value>"
    }
  ],
  "vpcConfig": {
    "securityGroupIds": [
      "<security-group-id>"
    ],
    "subnets": [
      "<subnet-id>"
    ],
    "vpcId": "<vpc-id>"
  },
  "badgeEnabled": "<badge-enabled>",
  "logsConfig": {
    "cloudWatchLogs": {
      "status": "<cloudwatch-logs-status>",
      "groupName": "<group-name>",

```

```

    "streamName": "<stream-name>"
  },
  "s3Logs": {
    "status": "<s3-logs-status>",
    "location": "<s3-logs-location>",
    "encryptionDisabled": "<s3-logs-encryption-disabled>"
  }
},
"fileSystemLocations": [
  {
    "type": "EFS",
    "location": "<EFS-DNS-name-1>:/<directory-path>",
    "mountPoint": "<mount-point>",
    "identifier": "<efs-identifier>",
    "mountOptions": "<efs-mount-options>"
  }
],
"buildBatchConfig": {
  "serviceRole": "<batch-service-role>",
  "combineArtifacts": <combine-artifacts>,
  "restrictions": {
    "maximumBuildsAllowed": <max-builds>,
    "computeTypesAllowed": [
      "<compute-type>"
    ],
    "fleetsAllowed": [
      "<fleet-name>"
    ]
  },
  "timeoutInMins": <batch-timeout>,
  "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
},
"concurrentBuildLimit": <concurrent-build-limit>
}

```

Sostituisci quanto segue:

name

Obbligatorio. Il nome di questo progetto di compilazione. Questo nome deve essere univoco per tutti i progetti di compilazione del tuo AWS account.

description

Facoltativo. La descrizione di questo progetto di compilazione.

source (origine)

Obbligatorio. Un [ProjectSource](#) oggetto che contiene informazioni sulle impostazioni del codice sorgente di questo progetto di build. Dopo aver aggiunto un oggetto `source`, è possibile aggiungere fino a più di 12 origini utilizzando la `source`. Queste impostazioni includono quanto segue:

fonte/ tipo

Obbligatorio. Il tipo di repository che contiene il codice sorgente da compilare. I valori validi includono:

- CODECOMMIT
- CODEPIPELINE
- GITHUB
- GITHUB_ENTERPRISE
- GITLAB
- GITLAB_SELF_MANAGED
- BITBUCKET
- S3
- NO_SOURCE

Se utilizzi `NO_SOURCE`, la specifica di compilazione non può essere un file perché il progetto non ha un'origine. Al contrario, devi utilizzare l'attributo `buildspec` per specificare una stringa in formato YAML per la tua specifica di compilazione. Per ulteriori informazioni, consulta [Crea un progetto di compilazione senza una fonte](#).

fonte/ ubicazione

Obbligatorio a meno che tu non lo `<source-type>` imposti. CODEPIPELINE La posizione del codice sorgente per il tipo di repository specificato.

- Per CodeCommit, l'URL del clone HTTPS del repository che contiene il codice sorgente e il file `buildspec` (ad esempio,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Per Amazon S3, il nome del bucket di input della build, seguito dal percorso e dal nome del file ZIP che contiene il codice sorgente e la `buildspec`. Per esempio:

- Per un file ZIP situato nella radice del bucket di input: `<bucket-name>/<object-name>.zip`
- Per un file ZIP che si trova in una sottocartella del bucket di input: `<bucket-name>/<subfolder-path>/<object-name>.zip`
- Per GitHub, l'URL del clone HTTPS del repository che contiene il codice sorgente e il file buildspec. L'URL deve contenere github.com. Devi connettere il tuo account al tuo account AWS GitHub. Per fare ciò, usa la CodeBuild console per creare un progetto di compilazione.
 - Selezionare Authorize application (Autorizza applicazione). (Dopo esserti connesso al tuo GitHub account, non è necessario completare la creazione del progetto di compilazione. Puoi chiudere la CodeBuild console.)
- Per GitHub Enterprise Server, l'URL clone HTTP o HTTPS del repository che contiene il codice sorgente e il file buildspec. È inoltre necessario collegare l' AWS account all'account Enterprise Server. GitHub A tale scopo, utilizzate la CodeBuild console per creare un progetto di compilazione.
 1. Crea un token di accesso personale in GitHub Enterprise Server.
 2. Copia questo token negli appunti in modo da poterlo utilizzare durante la CodeBuild creazione del progetto. Per ulteriori informazioni, consulta [Creazione di un token di accesso personale per la riga di comando sul sito Web](#) di GitHub Help.
 3. Quando usi la console per creare il tuo CodeBuild progetto, in Source, for Source provider, scegli GitHubEnterprise.
 4. Per Personal Access Token (Token di accesso personale), incollare il token che è stato copiato negli Appunti. Selezionare Save Token (Salva token). Il tuo CodeBuild account è ora collegato al tuo account GitHub Enterprise Server.
- GitLab Gestito automaticamente, l'URL clone HTTPS del repository che contiene il codice sorgente e il file buildspec. GitLab Tieni presente che, se lo utilizzi GitLab, l'URL deve contenere gitlab.com. Se utilizzi la GitLab gestione automatica, non è necessario che l'URL contenga gitlab.com. Devi collegare il tuo account al tuo account o al tuo AWS GitLab account autogestito. GitLab Per fare ciò, utilizzare la console CodeBuild per creare un progetto di compilazione.
 - Nel riquadro di navigazione Strumenti per sviluppatori, scegli Impostazioni, Connessioni e quindi Crea connessione. In questa pagina, crea una connessione GitLab o una connessione GitLab gestita automaticamente, quindi scegli Connetti a GitLab.
- Per Bitbucket, l'URL clone HTTPS del repository che contiene il codice sorgente e il file buildspec. L'URL deve contenere bitbucket.org. Devi anche connettere il tuo AWS account

al tuo account Bitbucket. Per fare ciò, usa la CodeBuild console per creare un progetto di compilazione.

1. Quando utilizzi la console per connetterti (o riconnetterti) a Bitbucket, nella pagina Confirm access to your account (Conferma l'accesso al tuo account) di Bitbucket, scegli Grant access (Concedi accesso). (Dopo esserti connesso al tuo account Bitbucket, non è necessario completare la creazione del progetto di build. Puoi chiudere la CodeBuild console.)
- Per AWS CodePipeline, non specificare un `location` valore persource. CodePipeline ignora questo valore perché quando create una pipeline in CodePipeline, specificate la posizione del codice sorgente nella fase Source della pipeline.

fonte/ gitCloneDepth

Facoltativo. La profondità della cronologia da scaricare. Il valore minimo è 0. Se il valore è pari a 0, maggiore di 25 o non fornito, viene scaricata la cronologia completa con ogni progetto di build. Se il tipo di origine è Amazon S3, questo valore non è supportato.

sorgente/ buildspec

Facoltativo. La definizione o il file delle specifiche di compilazione da utilizzare. Se tale valore non è fornito o è impostato su una stringa vuota, il codice di origine deve contenere un file `buildspec.yml` nella directory radice. Se questo valore è impostato, può essere una definizione buildspec in linea, il percorso di un file buildspec alternativo relativo alla directory principale della fonte primaria o il percorso di un bucket S3. Il bucket deve trovarsi nella stessa regione del progetto di compilazione. AWS Specificare il file buildspec utilizzando il relativo ARN (ad esempio, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Per ulteriori informazioni, consulta [Nome del file buildspec e posizione di storage](#).

fonte/ autenticazione

Contiene informazioni sulle impostazioni di autorizzazione per l'accesso CodeBuild al codice sorgente da compilare.

source/auth/ type

Obbligatorio. Il tipo di autorizzazione da utilizzare. I valori validi sono:

- OAUTH
- CODECONNECTIONS
- SECRETS_MANAGER

source/auth/ risorsa

Facoltativo. Il valore della risorsa che si applica al tipo di autorizzazione specificato. Può essere l'ARN di Secrets Manager o l' CodeConnections ARN.

fonte/ reportBuildStatus

Specifica se inviare al provider di origine lo stato di avvio e di completamento di una compilazione. Se lo imposti con un provider di origine diverso GitHub da GitHub Enterprise Server o Bitbucket, viene generato un. `invalidInputException`

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve disporre dell'accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

fonte/ buildStatusConfig

Contiene informazioni che definiscono il modo in cui il progetto di CodeBuild compilazione riporta lo stato della build al fornitore di origine. Questa opzione viene utilizzata solo quando il tipo di origine è `GITHUBGITHUB_ENTERPRISE`, `oBITBUCKET`.

contesto source//buildStatusConfig

Per le sorgenti Bitbucket, questo parametro viene utilizzato per il name parametro nello stato del commit di Bitbucket. Per GitHub le fonti, questo parametro viene utilizzato per il context parametro nello stato del commit. GitHub

Ad esempio, puoi fare in modo che context contenga il numero di build e il trigger del webhook utilizzando le variabili di CodeBuild ambiente:

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -  
${CODEBUILD_WEBHOOK_TRIGGER}
```

Il risultato è che il contesto appare così per la build #24 innescato da un evento di pull request del webhook:

```
AWS CodeBuild sample-project Build #24 - pr/8
```

source//targetUrl buildStatusConfig

Per le sorgenti Bitbucket, questo parametro viene utilizzato per il parametro nello stato del url commit di Bitbucket. Per GitHub le fonti, questo parametro viene utilizzato per il target_url parametro nello stato del commit. GitHub

Ad esempio, puoi impostare targetUrl to `https://aws.amazon.com/codebuild/<path to build>` e lo stato di commit verrà collegato a questo URL.

Puoi anche includere variabili di CodeBuild ambiente targetUrl per aggiungere ulteriori informazioni all'URL. Ad esempio, per aggiungere l'area di compilazione all'URL, imposta targetUrl su:

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Se l'area di compilazione è `us-east-2`, questa si espanderà a:

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

fonte/ gitSubmodulesConfig

Facoltativo. Informazioni sulla configurazione dei moduli secondari Git. Utilizzato solo con CodeCommit GitHub, GitHub Enterprise Server e Bitbucket.

source//gitSubmodulesConfigFetchSubModules

Impostare `fetchSubmodules` su `true` se desideri includere i moduli secondari Git nel repository. I moduli secondari Git inclusi devono essere configurati come HTTPS.

fonte/ InsecureSsl

Facoltativo. Utilizzato solo con GitHub Enterprise Server. Impostate questo valore su `true` per ignorare gli avvisi TLS durante la connessione al repository del progetto GitHub Enterprise Server. Il valore predefinito è `false`. `InsecureSsl` deve essere utilizzato solo a scopo di test. Non deve essere utilizzato in un ambiente di produzione.

source/ sourceIdentifier

Un identificatore definito dall'utente per l'origine del progetto. Facoltativo per la fonte principale. Obbligatorio per le fonti secondarie.

SecondarySources

Facoltativo. Una serie di [ProjectSource](#) oggetti che contengono informazioni sulle fonti secondarie per un progetto di compilazione. Puoi aggiungere fino a 12 fonti secondarie. Gli `secondarySources` oggetti utilizzano le stesse proprietà utilizzate dall'oggetto. In un oggetto di origine secondario, `sourceIdentifier` è obbligatorio.

secondarySourceVersions

Facoltativo. Un array di oggetti [ProjectSourceVersion](#). Se `secondarySourceVersions` vengono specificate a livello di compilazione, hanno la precedenza su questo.

SourceVersion

Facoltativo. La versione dell'input di compilazione da creare per questo progetto. Se non è specificato, viene utilizzata la versione più recente. Se specificato, deve essere uno tra:

- Per CodeCommit, l'ID di commit, il ramo o il tag Git da utilizzare.
- Ad GitHub esempio, l'ID di commit, l'ID della richiesta pull, il nome del ramo o il nome del tag che corrisponde alla versione del codice sorgente che desideri creare. Se si specifica l'ID di una richiesta pull, deve utilizzare il formato `pr/pull-request-ID` (ad esempio `pr/25`). Se un nome ramo è specificato, viene utilizzato l'ID commit HEAD del ramo. Se non è specificato, viene utilizzato l'ID commit HEAD del ramo predefinito.
- Ad esempio GitLab, l'ID di commit, l'ID della pull request, il nome del ramo, il nome del tag o il riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).
- Per Bitbucket, l'ID commit, il nome ramo o il nome del tag che corrisponde alla versione del codice sorgente che si desidera creare. Se un nome ramo è specificato, viene utilizzato l'ID commit HEAD del ramo. Se non è specificato, viene utilizzato l'ID commit HEAD del ramo predefinito.
- Per Amazon S3, l'ID versione dell'oggetto che rappresenta il file ZIP di input della compilazione da utilizzare.

Se `sourceVersion` è specificata a livello di compilazione, questa versione ha la precedenza rispetto a `sourceVersion` (a livello di progetto). Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

artefatti

Obbligatorio. Un [ProjectArtifacts](#) oggetto che contiene informazioni sulle impostazioni degli artefatti di output di questo progetto di build. Dopo aver aggiunto un oggetto `artifacts`, è possibile aggiungere fino a più di 12 artefatti utilizzando la . Queste impostazioni includono quanto segue:

artefatti/ tipo

Obbligatorio. Tipo di artefatto di output della build. I valori validi sono:

- `CODEPIPELINE`
- `NO_ARTIFACTS`
- `S3`

artefatti/ ubicazione

Utilizzato solo con il tipo di artefatto. `S3` Non utilizzato per altri tipi di artefatti.

Il nome del bucket di output che hai creato o identificato nei prerequisiti.

artefatti/ percorso

Utilizzato solo con il tipo di artefatto. `S3` Non utilizzato per altri tipi di artefatti.

Il percorso nel bucket di output in cui inserire il file o la cartella ZIP. Se non specificate un valore `perpath`, CodeBuild utilizza `namespaceType` (se specificato) e per `name` determinare il percorso e il nome del file ZIP o della cartella di output della build. Ad esempio, se si specifica `MyPath for path` e `MyArtifact.zip forname`, il percorso e il nome sarebbero `MyPath/MyArtifact.zip`.

artefatti/ NamespaceType

Utilizzato solo con il tipo di artefatto. `S3` Non utilizzato per altri tipi di artefatti.

Lo spazio dei nomi del file o della cartella ZIP di output della build. I valori validi includono `BUILD_ID` e `NONE`. Utilizzare `BUILD_ID` per inserire l'ID di compilazione nel percorso del file o della cartella ZIP di output di compilazione. In caso contrario, utilizzare `NONE`. Se non si specifica un valore per `namespaceType`, CodeBuild utilizza `path` (se specificato) e per `name` determinare il percorso e il nome del file ZIP o della cartella ZIP di output della build. Ad esempio, se si specifica `MyPath for pathnamespaceType`, `BUILD_ID for` e `MyArtifact.zip forname`, il percorso e il nome sarebbero `MyPath/build-ID/MyArtifact.zip`.

artifacts/name

Utilizzato solo con il tipo di `S3` artefatto. Non utilizzato per altri tipi di artefatti.

Il nome del file ZIP o della cartella di output della build all'interno di `location`. Ad esempio, se si specifica `MyPath` for `path` e `MyArtifact.zip` for `name`, il percorso e il nome sarebbero `MyPath/MyArtifact.zip`.

`artefatti/ overrideArtifactName`

Utilizzato solo con il tipo di artefatto S3. Non utilizzato per altri tipi di artefatti.

Facoltativo. Se impostato su `true`, il nome specificato nel `artifacts` blocco del file `buildspec` ha la precedenza. name Per ulteriori informazioni, consulta [Riferimento alle specifiche di costruzione per CodeBuild](#).

`artefatti/ imballaggio`

Utilizzato solo con il tipo di artefatto S3. Non utilizzato per altri tipi di artefatti.

Facoltativo. Specifica come impacchettare gli artefatti. I valori consentiti sono:

NONE

Crea una cartella che contenga gli artefatti della build. Si tratta del valore di default.

ZIP

Crea un file ZIP che contenga gli artefatti della build.

`secondaryArtifacts`

Facoltativo. Una serie di [ProjectArtifacts](#) oggetti che contengono informazioni sulle impostazioni degli artefatti secondari per un progetto di compilazione. È possibile aggiungere fino a 12 artefatti secondari. Il `secondaryArtifacts` utilizza molte delle stesse impostazioni utilizzate dall'oggetto .

`cache`

Obbligatorio. Un [ProjectCache](#) oggetto che contiene informazioni sulle impostazioni della cache di questo progetto di build. Per ulteriori informazioni, consulta [Compilazioni di cache](#).

`ambiente`

Obbligatorio. Un [ProjectEnvironment](#) oggetto che contiene informazioni sulle impostazioni dell'ambiente di compilazione di questo progetto. Queste impostazioni includono:

ambiente/ tipo

Obbligatorio. Il tipo di ambiente di compilazione. Per ulteriori informazioni, consulta [type](#) nel CodeBuild API Reference.

ambiente/ immagine

Obbligatorio. L'identificatore dell'immagine Docker utilizzato da questo ambiente di compilazione. In genere, questo identificatore è espresso come `image-name:tag`. Per esempio, nel repository Docker che CodeBuild utilizza per gestire le sue immagini Docker, questo potrebbe essere `aws/codebuild/standard:5.0`. In Docker Hub, `maven:3.3.9-jdk-8`. In Amazon ECR, `account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag`. Per ulteriori informazioni, consulta [Immagini Docker fornite da CodeBuild](#).

ambiente/ ComputeType

Obbligatorio. Specifica le risorse di calcolo utilizzate da questo ambiente di compilazione. Per ulteriori informazioni, consulta [ComputeType](#) nell'CodeBuild API Reference.

ambiente/ certificato

Facoltativo. L'ARN del bucket Amazon S3, il prefisso del percorso e la chiave oggetto che contiene il certificato con codifica PEM. La chiave dell'oggetto può essere solo il file `.pem` o un file `.zip` contenente il certificato con codifica PEM. Ad esempio, se il nome del bucket Amazon S3 è `<my-bucket>`, il prefisso del percorso è `<cert>` e il nome della chiave dell'oggetto è `<certificate.pem>`, i formati accettabili per sono o. `certificate <my-bucket/cert/certificate.pem>` `arn:aws:s3:::<my-bucket/cert/certificate.pem>`

Variabili di ambiente/ambiente

Facoltativo. Una matrice di [EnvironmentVariable](#) oggetti che contiene le variabili di ambiente che desideri specificare per questo ambiente di compilazione. Ogni variabile di ambiente è espressa come un oggetto che contiene un `name`, `value`, e `type`.

La console e AWS CLI gli utenti possono visualizzare tutte le variabili di ambiente. Se non avete dubbi sulla visibilità della variabile di ambiente, impostate `name` e `value` impostate `type` su `PLAINTEXT`.

Ti consigliamo di memorizzare le variabili di ambiente con valori sensibili, come un ID della chiave di AWS accesso, una chiave di accesso AWS segreta o una password, come parametro in Amazon EC2 Systems Manager Parameter Store o Gestione dei segreti AWS. Infatti `name`, per quel parametro memorizzato, imposta un identificatore CodeBuild a cui fare riferimento.

Se utilizzi Amazon EC2 Systems Manager Parameter Store, per `value`, imposta il nome del parametro così come memorizzato nel Parameter Store. Imposta `type` su `PARAMETER_STORE`. Utilizzando un parametro denominato `/CodeBuild/dockerLoginPassword` come esempio, imposta `name` su `suLOGIN_PASSWORD`. Imposta `value` su `/CodeBuild/dockerLoginPassword`. Imposta `type` su `PARAMETER_STORE`.

Important

Se utilizzi Amazon EC2 Systems Manager Parameter Store, ti consigliamo di memorizzare i parametri con nomi di parametri che iniziano con `/CodeBuild/` (ad esempio, `/CodeBuild/dockerLoginPassword`). Puoi utilizzare la CodeBuild console per creare un parametro in Amazon EC2 Systems Manager. Selezionare **Create a parameter** (Crea un parametro), quindi seguire le istruzioni nella finestra di dialogo. (In quella finestra di dialogo, per la chiave KMS, puoi specificare l'ARN di AWS KMS una chiave nel tuo account. Amazon EC2 Systems Manager utilizza questa chiave per crittografare il valore del parametro durante l'archiviazione e decrittografarlo durante il recupero.) Se usi la CodeBuild console per creare un parametro, la console avvia il nome del parametro con il `/CodeBuild/` momento in cui viene memorizzato. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.

Se il progetto di compilazione si riferisce a parametri memorizzati in Amazon EC2 Systems Manager Parameter Store, il ruolo di servizio del progetto di compilazione deve consentire `ssm:GetParameters`. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se invece hai scelto Existing service role (Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build si riferisce a parametri memorizzati in Amazon EC2 Systems Manager Parameter Store con nomi di parametri che non iniziano con `/CodeBuild/` e hai scelto Nuovo ruolo di servizio, devi aggiornare quel ruolo di servizio per consentire l'accesso ai nomi dei parametri che non iniziano con `/CodeBuild/`. Ciò dipende dal fatto che tale ruolo consente l'accesso solo a nomi di parametro che iniziano con `/CodeBuild/`.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i parametri nello `/CodeBuild/` spazio dei nomi in Amazon EC2 Systems Manager Parameter Store.

Le variabili di ambiente impostate sostituiscono quelle esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata `MY_VAR` con un valore di `my_value`, e si imposta una variabile di ambiente denominata `MY_VAR` con un valore di `other_value`, il valore `my_value` viene sostituito da `other_value`. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata `PATH` con un valore di `/usr/local/sbin:/usr/local/bin`, e si imposta una variabile di ambiente denominata `PATH` con un valore di `$PATH:/usr/share/ant/bin`, il valore di `/usr/local/sbin:/usr/local/bin` viene sostituito dal valore letterale `$PATH:/usr/share/ant/bin`.

Non impostare variabili di ambiente con nomi che iniziano con `CODEBUILD_`. Questo prefisso è riservato per l'uso interno .

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità.
- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità.
- Il valore della dichiarazione `buildspec` ha la minima priorità.

Se usi Secrets Manager, `forvalue`, imposta il nome del parametro come memorizzato in Secrets Manager. Imposta `type` su `SECRETS_MANAGER`. Utilizzando un segreto denominato `/CodeBuild/dockerLoginPassword` come esempio, impostate `name` su `LOGIN_PASSWORD`. Imposta `value` su `/CodeBuild/dockerLoginPassword`. Imposta `type` su `SECRETS_MANAGER`.

Important

Se utilizzi Secrets Manager, ti consigliamo di archiviare i segreti con nomi che iniziano con `/CodeBuild/` (ad esempio, `/CodeBuild/dockerLoginPassword`). Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager, il ruolo di servizio del progetto di build deve consentire l'azione `secretsmanager:GetSecretValue`. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo

progetto di build. Tuttavia, se invece hai scelto Existing service role (Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager con nomi segreti che non iniziano con /CodeBuild/ e hai scelto Nuovo ruolo di servizio, devi aggiornare il ruolo di servizio per consentire l'accesso ai nomi segreti che non iniziano con /CodeBuild/. Questo perché il ruolo di servizio consente l'accesso solo ai nomi segreti che iniziano con /CodeBuild/.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i segreti nello spazio dei /CodeBuild/ nomi in Secrets Manager.

ambiente/ RegistryCredential

Facoltativo. Un [RegistryCredential](#) oggetto che specifica le credenziali che forniscono l'accesso a un registro Docker privato.

Environment/registryCredential/ credential

Specifica l'ARN o il nome delle credenziali create utilizzando AWS Managed Services. Puoi utilizzare il nome delle credenziali solo se sono presenti nella regione corrente.

Environment/registryCredential/ CredentialProvider

L'unico valore valido è SECRETS_MANAGER.

Quando questo è impostato:

- `imagePullCredentials` deve essere impostato su `SERVICE_ROLE`.
- L'immagine non può essere un'immagine curata o un'immagine Amazon ECR.

ambiente/ Tipo imagePullCredentials

Facoltativo. Il tipo di credenziali CodeBuild utilizzate per estrarre le immagini nella build. Esistono due valori validi:

CODEBUILD

CODEBUILD specifica che CodeBuild utilizza le proprie credenziali. È necessario modificare la policy del repository Amazon ECR per considerare affidabile il responsabile del CodeBuild servizio.

SERVICE_ROLE

Specifica che CodeBuild utilizza il ruolo di servizio del progetto di compilazione.

Quando utilizzi un'immagine di registro condivisa tra account o privata, devi utilizzare le credenziali `SERVICE_ROLE`. Quando si utilizza un'immagine CodeBuild curata, è necessario utilizzare `CODEBUILD` le credenziali.

ambiente/ PrivilegedMode

Imposta su `true` solo se prevedi di utilizzare questo progetto di compilazione per creare immagini Docker. In caso contrario, tutte le compilazioni associate che tentano di interagire con il daemon Docker avranno esito negativo. È inoltre necessario avviare il daemon Docker in modo che le compilazioni possano interagire con esso. Un modo per fare ciò è inizializzare il daemon Docker nella fase `install` del file `buildspec` eseguendo i comandi di compilazione seguenti. Non eseguire questi comandi se è stata specificata un'immagine dell'ambiente di compilazione fornita da CodeBuild con supporto Docker.

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

serviceRole

Obbligatorio. L'ARN del ruolo di servizio CodeBuild viene utilizzato per interagire con i servizi per conto dell'utente (ad esempio, `arn:aws:iam::account-id:role/role-name`).

autoRetryLimit

Facoltativo. Il numero di tentativi automatici aggiuntivi dopo una compilazione non riuscita. Ad esempio, se il limite di tentativi automatici è impostato su 2, CodeBuild chiamerà l'`RetryBuildAPI` per riprovare automaticamente la build per un massimo di 2 volte aggiuntive.

timeoutInMinutes

Facoltativo. Il numero di minuti, compreso tra 5 e 2160 (36 ore), trascorsi i quali CodeBuild interrompe la compilazione se non è completa. Se non specificato, viene utilizzato il valore predefinito di 60. Per determinare se e quando una compilazione è stata CodeBuild interrotta a causa di un timeout, esegui il `batch-get-builds` comando. Per determinare se la compilazione si è interrotta, cercare nell'output un valore `buildStatus` di `FAILED`. Per determinare quando è scaduta la compilazione, cercare nell'output il valore `endTime` associato a un valore `phaseStatus` di `TIMED_OUT`.

queuedTimeoutInMinutes

Facoltativo. Il numero di minuti, compreso tra 5 e 480 (8 ore), dopo i quali CodeBuild interrompe la compilazione se è ancora in coda. Se non specificato, viene utilizzato il valore predefinito di 60.

encryptionKey

Facoltativo. L'alias o l'ARN utilizzato CodeBuild da per crittografare AWS KMS key l'output della build. Se si specifica un alias, utilizzare il formato `arn:aws:kms:region-ID:account-ID:key/key-ID` oppure, se un alias esiste, utilizzare il formato `alias/key-alias`. Se non specificato, viene utilizzata la chiave KMS AWS gestita per Amazon S3.

tags

Facoltativo. Una serie di oggetti [Tag](#) che forniscono i tag da associare a questo progetto di build. È possibile specificare fino a 50 tag. Questi tag possono essere utilizzati da qualsiasi AWS servizio che supporti i tag del progetto di CodeBuild compilazione. Ogni tag è espresso come un oggetto con a `key` e `value`.

vpcConfig

Facoltativo. Un [VpcConfig](#) oggetto che contiene informazioni sulla configurazione VPC per il tuo progetto. Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Queste proprietà includono:

vpclId

Obbligatorio. L'ID VPC che CodeBuild utilizza. Esegui questo comando per ottenere un elenco di tutti i VPC IDs nella tua regione:

```
aws ec2 describe-vpcs --region <region-ID>
```

sottoreti

Obbligatorio. Un array di sottoreti IDs che include le risorse utilizzate da CodeBuild. Esegui questo comando per ottenere questi IDs:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

securityGroupIds

Obbligatorio. Una serie di gruppi di sicurezza IDs utilizzati da CodeBuild per consentire l'accesso alle risorse nel VPC. Esegui questo comando per ottenere questi IDs:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

badgeEnabled

Facoltativo. Specifica se includere i badge di costruzione nel progetto. CodeBuild Imposta per `true` abilitare i badge di costruzione o altro. `false` Per ulteriori informazioni, consulta [Crea un esempio di badge con CodeBuild](#).

LogsConfig

Un [LogsConfig](#) oggetto che contiene informazioni su dove si trovano i log di questa build.

LogsConfig/ cloudWatchLogs

Un [CloudWatchLogsConfig](#) oggetto che contiene informazioni sull'invio di log a Logs. CloudWatch LogsConfig/ s3Logs

Un LogsConfig oggetto [S3](#) che contiene informazioni sull'invio di log ad Amazon S3.

fileSystemLocations

Facoltativo. Una serie di [ProjectFileSystemsLocation](#) oggetti che contiene informazioni sulla configurazione di Amazon EFS.

buildBatchConfig

Facoltativo. L'`buildBatchConfig` oggetto è una [ProjectBuildBatchConfig](#) struttura che contiene le informazioni di configurazione della build in batch per il progetto.

buildBatchConfig/ServiceRole

L'ARN del ruolo di servizio per il progetto di creazione in batch.

buildBatchConfig/CombineArtifacts

Un valore booleano che specifica se combinare gli elementi di compilazione per la creazione in batch in un'unica posizione degli artefatti.

buildBatchConfig/restrizioni/ maximumBuildsAllowed

Il numero massimo di build consentite.

buildBatchConfig/restrizioni/ computeTypesAllowed

Array di stringhe che specificano i tipi di calcolo consentiti per la compilazione batch. Vedi Tipi di [calcolo dell'ambiente di compilazione](#) per questi valori.

buildBatchConfig/restrictions/ FleetsAllowed

Un array di stringhe che specificano le flotte consentite per la compilazione in batch. Per ulteriori informazioni, consulta [Esegui build su flotte a capacità riservata](#).

buildBatchConfig/timeoutInMinutes

Il tempo massimo, in minuti, entro il quale la creazione in batch deve essere completata.

buildBatchConfig/batchReportMode

Specifica la modalità di invio dei rapporti sullo stato di compilazione al provider di origine per la compilazione in batch. I valori validi includono:

`REPORT_AGGREGATED_BATCH`

(Impostazione predefinita) Aggrega tutti gli stati di compilazione in un unico rapporto.

`REPORT_INDIVIDUAL_BUILDS`

Invia un rapporto distinto per ogni singola compilazione.

concurrentBuildLimit

Il numero massimo di compilazioni simultanee consentite per il progetto.

Le nuove compilazione vengono avviate solo se il numero corrente di compilazioni è minore o uguale a questo limite. Se il conteggio delle compilazioni correnti soddisfa questo limite, le nuove compilazioni vengono limitate e non vengono eseguite.

Creazione del progetto

Per creare il progetto, esegui nuovamente il [create-project](#) comando, passando il file JSON:

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

In caso di successo, la rappresentazione JSON di un oggetto [Project](#) viene visualizzata nell'output della console. Vedi la [sintassi di CreateProject risposta](#) per un esempio di questi dati.

A eccezione del nome del progetto di compilazione, è possibile modificare tutte le impostazioni del progetto di compilazione in un secondo momento. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#).

Per avviare l'esecuzione di una compilazione, consulta [Esecuzione di una compilazione \(AWS CLI\)](#).

Se il codice sorgente è archiviato in un GitHub repository e desideri CodeBuild ricostruirlo ogni volta che viene inserita una modifica al codice nel repository, consulta. [Avvio di esecuzioni di compilazione automaticamente \(AWS CLI\)](#)

Creazione di un progetto di compilazione (AWS SDKs)

Per informazioni sull'utilizzo AWS CodeBuild con, consulta la AWS SDKs. [AWS SDKs e riferimento agli strumenti](#)

Creazione di un progetto di compilazione (CloudFormation)

Per informazioni sull'utilizzo AWS CodeBuild con CloudFormation, consultate [il CloudFormation modello disponibile CodeBuild nella Guida per](#) l'AWS CloudFormation utente.

Creazione di una regola di notifica

È possibile utilizzare le regole di notifica per notificare agli utenti quando si verificano modifiche importanti, ad esempio compilazioni riuscite o non andate a buon fine. Le regole di notifica specificano sia gli eventi che l'argomento Amazon SNS utilizzato per inviare le notifiche. Per ulteriori informazioni, vedere [Cosa sono le notifiche?](#)

È possibile utilizzare la console o creare regole di notifica AWS CLI per. AWS CodeBuild

Per creare una regola di notifica (console)

1. Accedi a Console di gestione AWS e apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. Scegli Build, quindi Progetti build e poi scegli un progetto di build in cui desideri aggiungere le notifiche.
3. Nella pagina del progetto di compilazione, scegliere Notifica, quindi selezionare Crea regola di notifica. Puoi anche andare alla pagina Impostazioni per il progetto di build e scegliere Crea regola di notifica.
4. In Notification name (Nome notifica), immettere un nome per la regola.
5. In Tipo di dettaglio, scegli Basic se desideri che nella notifica siano EventBridge incluse solo le informazioni fornite ad Amazon. Scegli Completo se desideri includere le informazioni fornite ad Amazon EventBridge e le informazioni che potrebbero essere fornite da CodeBuild o dal gestore delle notifiche.

Per ulteriori informazioni, vedere [Informazioni sul contenuto delle notifiche e sulla sicurezza](#).

6. In Events that trigger notifications (Eventi che attivano le notifiche), selezionare gli eventi per i quali si desidera inviare notifiche. Per ulteriori informazioni, consulta l'argomento relativo agli [eventi per le regole di notifica sui progetti di compilazione](#).
7. In Targets (Destinazioni), procedere in uno dei seguenti modi:
 - Se hai già configurato una risorsa da utilizzare con le notifiche, in Scegli il tipo di destinazione, scegli Amazon Q Developer nelle applicazioni di chat (Slack) o l'argomento SNS. In Scegli destinazione, scegli il nome del client (per un client Slack configurato in Amazon Q Developer nelle applicazioni di chat) o l'Amazon Resource Name (ARN) dell'argomento Amazon SNS (per gli argomenti Amazon SNS già configurati con la politica richiesta per le notifiche).
 - Se non è stata configurata una risorsa da utilizzare con le notifiche, scegliere Create target (Crea destinazione), e quindi scegliere SNS topic (Argomento SNS). Fornire un nome per l'argomento dopo codestar-notifications-, quindi scegliere Create (Crea).

Note

- Se si crea l'argomento Amazon SNS come parte della creazione della regola di notifica, viene applicata la policy che consente alla funzionalità di notifica di pubblicare

eventi nell'argomento. L'utilizzo di un argomento creato per le regole di notifica consente di iscrivere solo gli utenti che si desidera ricevano le notifiche relative a questa risorsa.

- Non puoi creare un client Amazon Q Developer in chat come parte della creazione di una regola di notifica. Se scegli Amazon Q Developer nelle applicazioni di chat (Slack), vedrai un pulsante che ti indirizza a configurare un client in Amazon Q Developer nelle applicazioni di chat. Scegliendo questa opzione si apre la console delle applicazioni di chat di Amazon Q Developer. Per ulteriori informazioni, consulta [Configurare le integrazioni tra Notifications e Amazon Q Developer nelle applicazioni di chat](#).
- Se desideri utilizzare un argomento Amazon SNS esistente come obiettivo, devi aggiungere la politica richiesta per AWS CodeStar le notifiche oltre a qualsiasi altra politica che potrebbe esistere per quell'argomento. Per ulteriori informazioni, consulta l'argomento relativo alla [configurazione degli argomenti Amazon SNS per le notifiche](#) e le [informazioni sulla sicurezza e sui contenuti delle notifiche](#).

8. Per completare la creazione della regola, scegliere Invia.
9. È necessario iscrivere gli utenti all'argomento Amazon SNS relativo alla regola prima che possano ricevere notifiche. Per ulteriori informazioni, consulta [Sottoscrivere gli utenti agli argomenti di Amazon SNS relativi agli obiettivi](#). Puoi anche configurare l'integrazione tra le notifiche e Amazon Q Developer nelle applicazioni di chat per inviare notifiche alle chat room di Amazon Chime. Per ulteriori informazioni, consulta [Configurare l'integrazione tra Notifications e Amazon Q Developer nelle applicazioni di chat](#).

Per creare una regola di notifica (AWS CLI)

1. Da un terminale o dal prompt dei comandi, eseguire il comando `create-notification rule` per generare lo skeleton JSON:

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

È possibile assegnare al file un nome qualsiasi. In questo esempio, il file è denominato *rule.json*.

2. Aprire il file JSON in un editor di testo normale e modificarlo per includere la risorsa, i tipi di evento e la destinazione desiderata per la regola. L'esempio seguente mostra una regola di notifica denominata **MyNotificationRule** per un progetto di compilazione denominato

MyBuildProject in un AWS account con l'ID *123456789012*. Le notifiche vengono inviate con il tipo di dettaglio completo a un argomento di Amazon SNS denominato *codestar-notifications-MyNotificationTopic* quando le build hanno esito positivo:

```
{
  "Name": "MyNotificationRule",
  "EventTypeIds": [
    "codebuild-project-build-state-succeeded"
  ],
  "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",
  "Targets": [
    {
      "TargetType": "SNS",
      "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
    }
  ],
  "Status": "ENABLED",
  "DetailType": "FULL"
}
```

Salvare il file.

- Utilizzando il file appena modificato, dal terminale o dalla riga di comando, eseguire nuovamente il comando `create-notification-rule` per creare la regola di notifica:

```
aws codestarnotifications create-notification-rule --cli-input-json
file://rule.json
```

- In caso di esito positivo, il comando restituisce l'ARN della regola di notifica, simile al seguente:

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Modifica le impostazioni del progetto di compilazione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs modificare le impostazioni di un progetto di compilazione. AWS CLI

Se aggiungi report di test a un progetto di compilazione, assicurati che il tuo ruolo IAM disponga delle autorizzazioni descritte in [Autorizzazioni per il rapporto di prova](#).

Argomenti

- [Modifica delle impostazioni di un progetto di compilazione \(console\)](#)
- [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#)
- [Modifica delle impostazioni di un progetto di compilazione \(AWS SDKs\)](#)

Modifica delle impostazioni di un progetto di compilazione (console)

Per modificare le impostazioni per un progetto di compilazione, esegui la seguente procedura:

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Esegui una delle seguenti operazioni:
 - Selezionare il collegamento del progetto di compilazione da modificare, quindi selezionare Build details (Dettagli compilazione).
 - Selezionare il pulsante di opzione accanto al progetto di compilazione da modificare, selezionare View details (Visualizza dettagli) e Build details (Dettagli compilazione).

È possibile modificare le seguenti sezioni:

Sections

- [Configurazione del progetto](#)
- [Origine](#)
- [Ambiente](#)
- [Buildspec](#)
- [Configurazione Batch](#)

- [Artifacts](#)
- [Log](#)

Configurazione del progetto

Nella sezione Configurazione del progetto, scegli Modifica. Quando le modifiche sono complete, scegli Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà.

Descrizione

Immettete una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.

Costruisci un badge

Selezionare Enable build badge (Abilita badge di compilazione) per rendere visibile e integrabile lo stato della compilazione del progetto. Per ulteriori informazioni, consulta [Creazione di esempi di badge](#).

Note

Il badge Build non si applica se il tuo provider di origine è Amazon S3.

Abilita il limite di compilazione simultanea

Se desideri limitare il numero di build simultanee per questo progetto, esegui i seguenti passaggi:

1. Seleziona Limita il numero di build simultanee che questo progetto può avviare.
2. In Limite di build simultanee, inserisci il numero massimo di build simultanee consentite per questo progetto. Questo limite non può essere superiore al limite di creazione simultanea impostato per l'account. Se si tenta di inserire un numero superiore al limite dell'account, viene visualizzato un messaggio di errore.

Le nuove compilazione vengono avviate solo se il numero corrente di compilazioni è minore o uguale a questo limite. Se il conteggio delle compilazioni correnti soddisfa questo limite, le nuove compilazioni vengono limitate e non vengono eseguite.

Abilita l'accesso pubblico alla build

Per rendere disponibili al pubblico i risultati della compilazione del progetto, inclusi gli utenti senza accesso a un AWS account, seleziona Abilita l'accesso pubblico alla build e conferma che desideri rendere pubblici i risultati della build. Le seguenti proprietà vengono utilizzate per i progetti di compilazione pubblica:

Ruolo del servizio di costruzione pubblico

Seleziona Nuovo ruolo di servizio se desideri CodeBuild creare un nuovo ruolo di servizio per te o Ruolo di servizio esistente se desideri utilizzare un ruolo di servizio esistente.

Il ruolo del servizio di compilazione pubblico CodeBuild consente di leggere CloudWatch i log e scaricare gli artefatti di Amazon S3 per le build del progetto. Questo è necessario per rendere disponibili al pubblico i log di compilazione e gli artefatti del progetto.

Ruolo di servizio

Immettere il nome del nuovo ruolo di servizio o di un ruolo di servizio esistente.

Per rendere privati i risultati di compilazione del progetto, deseleziona Abilita l'accesso pubblico alla build.

Per ulteriori informazioni, consulta [Get public build project URLs](#).

Warning

Quando rendi pubblici i risultati di compilazione del progetto, tieni presente quanto segue:

- Tutti i risultati di compilazione, i log e gli artefatti di un progetto, incluse le build eseguite quando il progetto era privato, sono disponibili al pubblico.
- Tutti i log e gli artefatti di compilazione sono disponibili al pubblico. Le variabili di ambiente, il codice sorgente e altre informazioni sensibili potrebbero essere state inserite nei log e negli artefatti di compilazione. È necessario prestare attenzione alle informazioni che vengono emesse nei log di compilazione. Alcune best practice sono:
 - Non memorizzate valori sensibili, in particolare gli ID delle chiavi di AWS accesso e le chiavi di accesso segrete, nelle variabili di ambiente. Ti consigliamo di utilizzare un Amazon EC2 Systems Manager Parameter Store Gestione dei segreti AWS o di archiviare valori sensibili.
 - Limita [Procedure consigliate per l'utilizzo dei webhook](#) le entità che possono attivare una build e non memorizzare le specifiche della build nel progetto stesso, per assicurarti che i tuoi webhook siano il più sicuri possibile.

- Un utente malintenzionato può utilizzare build pubbliche per distribuire artefatti dannosi. Consigliamo agli amministratori del progetto di esaminare tutte le pull request per verificare che la pull request sia una modifica legittima. Ti consigliamo inoltre di convalidare tutti gli artefatti con i relativi checksum per assicurarti che vengano scaricati gli artefatti corretti.

Informazioni aggiuntive

Per Tag, inserisci il nome e il valore di tutti i tag che desideri vengano utilizzati dai servizi di supporto. AWS Utilizzare Add row (Aggiungi riga) per aggiungere un tag. Puoi aggiungere fino a 50 tag.

Origine

Nella sezione Sorgente, scegli Modifica. Una volta completate le modifiche, scegli Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà:

Provider di origine

Scegli il tipo di provider del codice sorgente. Utilizza i seguenti elenchi per effettuare le selezioni appropriate per il tuo provider di origine:

Note

CodeBuild non supporta Bitbucket Server.

Amazon S3

Bucket

Scegli il nome del bucket di input che contiene il codice sorgente.

Chiave oggetto S3 o cartella S3

Inserisci il nome del file ZIP o il percorso della cartella che contiene il codice sorgente. Inserisci una barra (/) per scaricare tutti i contenuti del bucket S3.

Versione sorgente

Immettete l'ID di versione dell'oggetto che rappresenta la build del file di input. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

CodeCommit

Repository

Scegli il repository che desideri utilizzare.

Tipo di riferimento

Scegli Branch, Git tag o Commit ID per specificare la versione del codice sorgente. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Scegli di creare un clone poco profondo con una cronologia troncata al numero di commit specificato. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Bitbucket

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli CodeConnections la OAuthpassword dell'app o il token di accesso personale a cui connetterti CodeBuild.

Connessione

Seleziona una connessione Bitbucket o un segreto Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio account Bitbucket o Archivio pubblico e inserisci l'URL del repository.

Versione sorgente

Inserisci un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#)

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto Status, inserisci il valore da utilizzare per il name parametro nello stato del commit di Bitbucket. Per ulteriori informazioni, consulta [compilazione](#) nella documentazione di riferimento dell'API Bitbucket.

Per Target URL, inserisci il valore da utilizzare per il url parametro nello stato del commit di Bitbucket. Per ulteriori informazioni, consulta [compilazione](#) nella documentazione di riferimento dell'API Bitbucket.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

Negli eventi del webhook di origine primaria, seleziona Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository se desideri creare il codice sorgente ogni volta che viene inviata una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta. [Eventi webhook Bitbucket](#)

GitHub

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli GitHub App o Token di accesso personale a cui connetterti CodeBuild. OAuth

Connessione

Seleziona una GitHub connessione o un segreto di Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio GitHub account, Public repository o GitHub scoped webhook e inserisci l'URL del repository.

Versione sorgente

Inserisci un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#)

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto Status, immettete il valore da utilizzare per il context parametro nel GitHub commit status. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Per Target URL, inserisci il valore da utilizzare per il `target_url` parametro nello stato del GitHub commit. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

Negli eventi del webhook di origine primaria, seleziona Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository se desideri creare il codice sorgente ogni volta che viene inviata una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta [GitHub eventi webhook](#)

GitHub Enterprise Server

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

Scegli il CodeConnectionsnostro token di accesso personale a cui connetterti CodeBuild.

Connessione

Seleziona una connessione GitHub Enterprise o un segreto Secrets Manager per connetterti tramite il tipo di connessione specificato.

Repository

Scegli Repository nel mio account GitHub Enterprise o GitHub Enterprise scoped webhook e inserisci l'URL del repository.

Versione sorgente

Inserisci una pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

 Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Sottomoduli Git

Seleziona Use Git submodules (Utilizza i sottomoduli Git) se vuoi includere i sottomoduli Git nel tuo archivio.

Stato della build

Seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Per il contesto Status, immettete il valore da utilizzare per il context parametro nel GitHub commit status. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Per Target URL, inserisci il valore da utilizzare per il target_url parametro nello stato del GitHub commit. Per ulteriori informazioni, consulta [Creare uno stato di commit](#) nella guida per GitHub sviluppatori.

Lo stato di una build attivata da un webhook viene sempre segnalato al provider di origine. Per far sì che lo stato di una build avviata dalla console o di una chiamata API venga segnalato al provider di origine, è necessario selezionare questa impostazione.

Se le build del progetto vengono attivate da un webhook, è necessario inviare un nuovo commit al repository affinché una modifica a questa impostazione abbia effetto.

SSL non sicuro

Seleziona Abilita SSL non sicuro per ignorare gli avvisi SSL durante la connessione al repository del tuo progetto Enterprise. GitHub

In Primary source webhook events, seleziona Ricostruisci ogni volta che viene inserita una modifica al codice in questo repository se desideri creare il codice sorgente ogni volta che viene inserita una modifica CodeBuild al codice in questo repository. Per ulteriori informazioni sui webhook e sui gruppi di filtri, consulta. [GitHub eventi webhook](#)

GitLab

Credenziali

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

CodeConnectionsviene utilizzato per connettersi GitLab a CodeBuild.

Connessione

Seleziona una GitLab connessione tramite la quale connetterti CodeConnections.

Repository

Scegli il repository che desideri utilizzare.

Versione sorgente

Inserisci un ID di pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Stato della build

Seleziona Segnala lo stato della compilazione al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

GitLab Self Managed

Credenziale

Scegli Credenziale di origine predefinita o Credenziale di origine personalizzata e segui le istruzioni per gestire la credenziale di origine predefinita o personalizzare la credenziale di origine.

Tipo di connessione

CodeConnectionsviene utilizzato per connettere GitLab Self Managed a CodeBuild.

Connessione

Seleziona una connessione GitLab Self Managed tramite la quale connetterti CodeConnections.

Repository

Scegli il repository che desideri utilizzare.

Versione sorgente

Inserisci un ID di pull request, un branch, un ID di commit, un tag o un riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

 Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come 811dd1ba1aba14473856cee38308caed7190c0d o5392f7. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

Profondità del clone Git

Seleziona Git clone depth per creare un clone superficiale con una cronologia troncata al numero specifico di commit. Se si desidera un clone completo, selezionare Full (Completo).

Stato della build

Seleziona Segnala lo stato della compilazione al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Ambiente

Nella sezione Ambiente, scegli Modifica. Una volta completate le modifiche, scegli Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà:

Modello di provisioning

Per modificare il modello di provisioning, scegli Cambia modello di provisioning ed esegui una delle seguenti operazioni:

- Per utilizzare flotte on-demand gestite da AWS CodeBuild, scegli On-demand. Con le flotte on-demand, CodeBuild fornisce elaborazione per le tue build. Le macchine vengono distrutte al termine della costruzione. Le flotte on-demand sono completamente gestite e includono funzionalità di scalabilità automatica per gestire i picchi di domanda.
- Per utilizzare flotte a capacità riservata gestite da AWS CodeBuild, scegli Capacità riservata, quindi seleziona il nome di una flotta. Con le flotte a capacità riservata, puoi configurare un set

di istanze dedicate per il tuo ambiente di costruzione. Queste macchine rimangono inattive, pronte per l'elaborazione immediata di build o test e riducono la durata di costruzione. Con flotte a capacità riservata, le macchine sono sempre in funzione e continueranno a sostenere costi per tutto il tempo in cui vengono rifornite.

Per informazioni, consulta [Run si basa su flotte a capacità riservata](#).

Immagine dell'ambiente

Per modificare l'immagine di costruzione, scegliete Sostituisci immagine ed effettuate una delle seguenti operazioni:

- Per utilizzare un'immagine Docker gestita da AWS CodeBuild, scegli Immagine gestita, quindi effettua le selezioni tra Sistema operativo, Runtime (s), Image e Image version. Se disponibile, eseguire una selezione da Environment type (Tipo ambiente).
- Per utilizzare un'altra immagine Docker, selezionare Custom image (Immagine personalizzata). Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Se si sceglie Other registry (Altro registro), fin External registry URL (URL registro esterno), inserire il nome e il tag dell'immagine Docker in Docker Hub, utilizzando il formato *docker repository/docker image name*. Se scegli Amazon ECR, utilizza il repository Amazon ECR e l'immagine Amazon ECR per scegliere l'immagine Docker nel tuo account. AWS
- Per utilizzare un'immagine Docker privata, scegli Immagine personalizzata. Per il tipo di ambiente, scegli ARM, Linux, Linux GPU o Windows. Per Image registry (Registro immagine), selezionare Other registry (Altro registro), quindi immettere l'ARN delle credenziali per l'immagine Docker privata. Le credenziali devono essere create da Secrets Manager. Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS .

Note

CodeBuild sostituisce le immagini Docker ENTRYPOINT personalizzate.

Ruolo di servizio

Esegui una delle seguenti operazioni:

- Se non disponi di un ruolo CodeBuild di servizio, scegli Nuovo ruolo di servizio. In Nome ruolo, inserisci un nome per il nuovo ruolo.

- Se hai un ruolo CodeBuild di servizio, scegli Ruolo di servizio esistente. In Role ARN, scegli il ruolo di servizio.

Note

Quando si utilizza la console per creare un progetto di compilazione, è possibile creare contemporaneamente un ruolo di CodeBuild servizio. Per impostazione predefinita, il ruolo funziona solo con tale progetto di compilazione. Se si utilizza la console per associare questo ruolo del servizio con un altro progetto di compilazione, il ruolo viene aggiornato per funzionare con l'altro progetto. Un ruolo del servizio può funzionare con un massimo di 10 progetti di compilazione.

Configurazione aggiuntiva

Timeout

Specificate un valore, compreso tra 5 minuti e 36 ore, dopo il quale CodeBuild interrompe la compilazione se non è completa. Se hours (ore) e minutes (minuti) vengono lasciati vuoti, viene utilizzato il valore predefinito di 60 minuti.

Privilegiato

Seleziona Abilita questo flag se desideri creare immagini Docker o desideri che le tue build ottengano privilegi elevati. solo se prevedi di utilizzare questo progetto di compilazione per creare immagini Docker. In caso contrario, tutte le compilazioni associate che tentano di interagire con il daemon Docker avranno esito negativo. È inoltre necessario avviare il daemon Docker in modo che le compilazioni possano interagire con esso. Un modo per fare ciò è inizializzare il daemon Docker nella fase `install` delle specifiche di compilazione eseguendo i comandi di compilazione seguenti. Non eseguire questi comandi se hai scelto un'immagine dell'ambiente di compilazione CodeBuild fornita dal supporto Docker.

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Se vuoi CodeBuild lavorare con il tuo VPC:

- Per VPC, scegli l'ID VPC che utilizza. CodeBuild
- Per le sottoreti VPC, scegli le sottoreti che includono le risorse che utilizza. CodeBuild
- Per i gruppi di sicurezza VPC, scegli i gruppi di sicurezza da CodeBuild utilizzare per consentire l'accesso alle risorse in VPCs

Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Calcolo

Scegli una delle opzioni disponibili.

Credenziali di registro

Specificare una credenziale di registro quando il progetto è configurato con un'immagine di registro non privata.

Note

Questa credenziale verrà utilizzata solo se le immagini vengono sostituite con quelle dei registri privati.

Variabili di ambiente

Immettete il nome e il valore, quindi scegliete il tipo di ogni variabile di ambiente da utilizzare per le build.

Note

CodeBuild imposta automaticamente la variabile di ambiente per la tua AWS regione. È necessario impostare le seguenti variabili di ambiente se non le hai aggiunte al tuo buildspec.yml:

- AWS_ACCOUNT_ID
- IMAGE_REPO_NAME
- IMAGE_TAG

La console e AWS CLI gli utenti possono visualizzare le variabili di ambiente. Se la visibilità della variabile di ambiente non rappresenta un problema, configurare i campi Name (Nome) e Value (Valore), quindi impostare Type (Tipo) su Plaintext (Testo normale).

Ti consigliamo di memorizzare una variabile di ambiente con un valore sensibile, come un ID di chiave di AWS accesso, una chiave di accesso AWS segreta o una password come parametro in Amazon EC2 Systems Manager Parameter Store Gestione dei segreti AWS o.

Se utilizzi Amazon EC2 Systems Manager Parameter Store, per Tipo scegli Parameter. Per Nome, inserisci un identificativo a cui CodeBuild fare riferimento. Per Value, inserisci il nome del parametro memorizzato in Amazon EC2 Systems Manager Parameter Store. Utilizzando un parametro esemplificativo denominato /CodeBuild/dockerLoginPassword, per Type (Tipo) selezionare Parameter (Parametro). In Nome, inserisci LOGIN_PASSWORD. In Valore, specifica /CodeBuild/dockerLoginPassword.

Important

Se utilizzi Amazon EC2 Systems Manager Parameter Store, ti consigliamo di archiviare i parametri con nomi di parametri che iniziano /CodeBuild/ con (ad esempio /CodeBuild/dockerLoginPassword). Puoi utilizzare la CodeBuild console per creare un parametro in Amazon EC2 Systems Manager. Selezionare Create a parameter (Crea un parametro), quindi seguire le istruzioni nella finestra di dialogo. (In quella finestra di dialogo, per la chiave KMS, puoi specificare l'ARN di AWS KMS una chiave nel tuo account. Amazon EC2 Systems Manager utilizza questa chiave per crittografare il valore del parametro durante l'archiviazione e decrittografarlo durante il recupero.) Se usi la CodeBuild console per creare un parametro, la console avvia il nome del parametro con /CodeBuild/ il momento in cui viene memorizzato. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.

Se il progetto di compilazione si riferisce a parametri archiviati in Amazon EC2 Systems Manager Parameter Store, il ruolo di servizio del progetto di compilazione

deve consentire `ssm:GetParameters` l'azione. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se invece hai scelto Existing service role (Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build si riferisce a parametri archiviati in Amazon EC2 Systems Manager Parameter Store con nomi di parametri che non iniziano `/CodeBuild/` con e hai scelto Nuovo ruolo di servizio, devi aggiornare quel ruolo di servizio per consentire l'accesso ai nomi dei parametri che non iniziano `/CodeBuild/` con. Ciò dipende dal fatto che tale ruolo consente l'accesso solo a nomi di parametro che iniziano con `/CodeBuild/`.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i parametri nello spazio dei `/CodeBuild/` nomi in Amazon EC2 Systems Manager Parameter Store.

Le variabili di ambiente impostate sostituiscono quelle esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata `MY_VAR` con un valore di `my_value`, e si imposta una variabile di ambiente denominata `MY_VAR` con un valore di `other_value`, il valore `my_value` viene sostituito da `other_value`. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata `PATH` con un valore di `/usr/local/sbin:/usr/local/bin`, e si imposta una variabile di ambiente denominata `PATH` con un valore di `$PATH:/usr/share/ant/bin`, il valore di `/usr/local/sbin:/usr/local/bin` viene sostituito dal valore letterale `$PATH:/usr/share/ant/bin`.

Non impostare variabili di ambiente con nomi che iniziano con `CODEBUILD_`. Questo prefisso è riservato per l'uso interno .

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità.
- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità.
- Il valore della dichiarazione `buildspec` ha la minima priorità.

Se usi Secrets Manager, per Tipo, scegli Secrets Manager. Per Nome, inserite un identificatore CodeBuild a cui fare riferimento. Per Valore, immettere un `reference-`

key utilizzando il modello `secret-id:json-key:version-stage:version-id`. Per informazioni, consulta [Secrets Manager reference-key in the buildspec file](#).

Important

Se utilizzi Secrets Manager, ti consigliamo di archiviare i segreti con nomi che iniziano con `/CodeBuild/` (ad esempio, `/CodeBuild/dockerLoginPassword`). Per ulteriori informazioni, consulta [Che cos'è Gestione dei segreti AWS?](#) nella Guida per l'utente di Gestione dei segreti AWS.

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager, il ruolo di servizio del progetto di build deve consentire l'azione `secretsmanager:GetSecretValue`. Se in precedenza hai scelto Nuovo ruolo di servizio, CodeBuild include questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se invece hai scelto Existing service role (Ruolo del servizio esistente), devi includere questa operazione al ruolo del servizio separatamente.

Se il tuo progetto di build fa riferimento a segreti archiviati in Secrets Manager con nomi segreti che non iniziano con `/CodeBuild/` e hai scelto Nuovo ruolo di servizio, devi aggiornare il ruolo di servizio per consentire l'accesso ai nomi segreti che non iniziano con `/CodeBuild/`. Questo perché il ruolo di servizio consente l'accesso solo ai nomi segreti che iniziano con `/CodeBuild/`.

Se scegli Nuovo ruolo di servizio, il ruolo di servizio include l'autorizzazione a decrittografare tutti i segreti nello spazio dei `/CodeBuild/` nomi in Secrets Manager.

Buildspec

Nella sezione Buildspec, scegli Modifica. Quando le modifiche sono complete, scegli Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà:

Specifiche di costruzione

Esegui una delle seguenti operazioni:

- Se il codice sorgente include un file di specifiche di compilazione, selezionare Use a buildspec file (Utilizza un file buildspec). Per impostazione predefinita, CodeBuild cerca un file denominato `buildspec.yml` nella directory principale del codice sorgente. Se il tuo file buildspec utilizza

un nome o una posizione diversi, inserisci il percorso dalla radice di origine in `Buildspec name` (ad esempio, `o. buildspec-two.yml configuration/buildspec.yml`). Se il file `buildspec` si trova in un bucket S3, deve trovarsi nella stessa regione del progetto di build. AWS Specificate il file `buildspec` utilizzando il relativo ARN (ad esempio,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`

- Se il codice sorgente non include un file di specifiche di compilazione o se desideri eseguire comandi di compilazione diversi da quelli specificati per la fase `build` del file `buildspec.yml` nella directory radice del codice sorgente, scegli `Insert build commands` (Inserisci comandi di compilazione). Per `Build commands` (Comandi di compilazione), immetti i comandi che desideri eseguire nella fase `build`. In caso di più comandi, separa ogni comando con `&&` (ad esempio `mvn test && mvn package`). Per eseguire comandi in altre fasi, o se disponi di un lungo elenco di comandi per la `build` fase, aggiungi un `buildspec.yml` file alla directory principale del codice sorgente, aggiungi i comandi al file e quindi scegli `Usa il buildspec.yml nella directory principale del codice sorgente`.

Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

Configurazione Batch

Nella sezione `Configurazione Batch`, scegli `Modifica`. Una volta completate le modifiche, scegli `Aggiorna configurazione` per salvare la nuova configurazione. Per ulteriori informazioni, consulta [Esegui le build in batch](#).

È possibile modificare le seguenti proprietà:

Ruolo del servizio Batch

Fornisce il ruolo di servizio per le build in batch.

Seleziona una delle seguenti opzioni:

- Se non disponi di un ruolo di servizio batch, scegli `Nuovo ruolo di servizio`. In `Ruolo di servizio`, inserisci un nome per il nuovo ruolo.
- Se disponi di un ruolo di servizio batch, scegli `Ruolo di servizio esistente`. In `Ruolo di servizio`, scegli il ruolo di servizio.

Le compilazioni in batch introducono un nuovo ruolo di sicurezza nella configurazione in batch. Questo nuovo ruolo è obbligatorio in quanto CodeBuild deve poter chiamare `StartBuildStopBuild`, e `RetryBuild` le azioni per tuo conto per eseguire le build come

parte di un batch. I clienti devono utilizzare un nuovo ruolo e non lo stesso ruolo che usano nella compilazione, per due motivi:

- Dare al ruolo di compilazione le autorizzazioni `StartBuild`, `StopBuild` e `RetryBuild` consentirebbe a una singola compilazione di avviare più compilazioni tramite `buildspec`.
- CodeBuild le build in batch forniscono restrizioni che limitano il numero di build e i tipi di calcolo che possono essere utilizzati per le build del batch. Se il ruolo di compilazione dispone di queste autorizzazioni, è possibile che le compilazioni stesse possano ignorare queste restrizioni.

Tipi di calcolo consentiti per batch

Seleziona i tipi di calcolo consentiti per il batch. Seleziona tutte le risposte pertinenti.

Flotte consentite per lotto

Seleziona le flotte consentite per il batch. Seleziona tutte le risposte pertinenti.

Numero massimo di build consentite in batch

Immettere il numero massimo di build consentite nel batch. Se un batch supera questo limite, il batch avrà esito negativo.

Timeout del batch

Immettere il periodo di tempo massimo per il completamento della creazione del batch.

Combina artefatti

Seleziona Combina tutti gli artefatti del batch in un'unica posizione per combinare tutti gli artefatti del batch in un'unica posizione.

Modalità report Batch

Seleziona la modalità di report sullo stato della build desiderata per le compilazioni in batch.

Note

Questo campo è disponibile solo quando l'origine del progetto è Bitbucket o GitHub Enterprise e in Origine è selezionato Segnala lo stato della build al fornitore di origine quando le build iniziano e finiscono. GitHub

Building aggregate

Seleziona questa opzione per combinare gli stati di tutte le build del batch in un unico rapporto sullo stato.

Building individuali

Seleziona questa opzione per far sì che gli stati di compilazione di tutte le build del batch vengano riportati separatamente.

Artifacts

Nella sezione Artefatti, scegli Modifica. Una volta completate le modifiche, scegliete Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà:

Tipo

Esegui una delle seguenti operazioni:

- Se non si desidera creare alcun artefatto di output di compilazione, scegliere No artifacts (Nessun artefatto). Questa operazione potrebbe essere utile se esegui solo test di compilazione o desideri inviare un'immagine Docker a un repository Amazon ECR.
- Per archiviare l'output della build in un bucket S3, scegli Amazon S3, quindi procedi come segue:
 - Se desideri utilizzare il tuo nome di progetto per la cartella o il file ZIP di output di compilazione, lascia vuoto il campo Name (Nome). In caso contrario, digitare il nome. Se desideri eseguire l'output di un file ZIP e vuoi che il file ZIP abbia un'estensione di file, assicurati di includerla dopo il nome del file ZIP.
 - Selezionare Enable semantic versioning (Abilita funzione Versioni multiple semantica) se si desidera che un nome specificato nel file buildspec sostituisca qualsiasi nome specificato nella console. Il nome del file di specifiche di compilazione viene calcolato in fase di compilazione e utilizza il linguaggio di comando Shell. Ad esempio, è possibile aggiungere una data e un'ora al nome dell'artefatto in modo che sia sempre univoco. I nomi di artefatto univoci impediscono che gli artefatti vengano sovrascritti. Per ulteriori informazioni, consulta [Sintassi buildspec](#).
- Per Bucket name (Nome bucket) selezionare il nome del bucket di output.

- Se hai scelto Insert build commands (Inserisci comandi di compilazione) in una fase precedente di questa procedura, quindi per Output files (File di output), immetti le posizioni dei file di compilazione che desideri inserire nella cartella o nel file ZIP dell'output di compilazione. Per più posizioni, separarne ognuna con una virgola (per esempio, `appspec.yml`, `target/my-app.jar`). Per ulteriori informazioni, consultare la descrizione di files in [Sintassi buildspec](#).
- Se desideri crittografare gli artefatti di compilazione, seleziona Remove artifacts encryption (Rimuovi crittografia artefatti).

Per ogni insieme secondario di artefatti desiderato:

1. Per Artifact identifier (Identificatore di artefatto), inserire un valore inferiore a 128 caratteri e che contiene solo caratteri alfanumerici e trattini bassi.
2. Scegliere Add artifact (Aggiungi artefatto).
3. Seguire le fasi precedenti configurare gli artefatti secondari.
4. Selezionare Save artifact (Salva elemento).

Configurazione aggiuntiva

Chiave di crittografia

Esegui una delle seguenti operazioni:

- Per utilizzare Chiave gestita da AWS Amazon S3 nel tuo account per crittografare gli artefatti di output della build, lascia vuota la chiave di crittografia. Questa è l'impostazione predefinita.
- Per utilizzare una chiave gestita dal cliente per crittografare gli artefatti di output della build, in Chiave di crittografia, inserisci l'ARN della chiave gestita dal cliente. Utilizzare il formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo di cache

In Cache type (Tipo di cache), procedere in uno dei modi seguenti:

- Se non si desidera utilizzare una cache, scegliere No cache (Nessuna cache).
- Se desideri utilizzare una cache Amazon S3, scegli Amazon S3 e procedi come segue:
 - Per Bucket, selezionare il nome del bucket S3 in cui è archiviata la cache.

- (Facoltativo) Per il prefisso del percorso della cache, inserisci un prefisso del percorso Amazon S3. Il valore Cache path prefix (Prefisso percorso cache) è simile a un nome di directory. Consente di archiviare la cache sotto la stessa directory in un bucket.

 Important

Non aggiungere una barra finale (/) alla fine del prefisso del percorso.

- Se si desidera utilizzare una cache locale, selezionare Local (Locale), quindi scegliere una o più modalità cache locali.

 Note

La modalità Cache di livello Docker è disponibile solo per Linux. Se la scegli, il progetto deve essere eseguito in modalità privilegiata.

L'utilizzo di una cache consente di risparmiare tempo di compilazione perché alcune parti riutilizzabili dell'ambiente di compilazione vengono memorizzate nella cache e utilizzate in compilazioni diverse. Per informazioni su come specificare una cache nel file di specifiche di compilazione, consulta [Sintassi buildspec](#). Per ulteriori informazioni sul caching, consulta [Creazioni di cache per migliorare le prestazioni](#).

Log

Nella sezione Registri, scegli Modifica. Una volta completate le modifiche, scegli Aggiorna configurazione per salvare la nuova configurazione.

È possibile modificare le seguenti proprietà:

Scegli i log che desideri creare. Puoi creare Amazon CloudWatch Logs, Amazon S3 log o entrambi.

CloudWatch

Se desideri i log di Amazon CloudWatch Logs:

CloudWatch logs

Seleziona i logCloudWatch .

Group name (Nome gruppo)

Inserisci il nome del tuo gruppo di CloudWatch log Amazon Logs.

Nome dello stream

Inserisci il nome del tuo flusso di CloudWatch log di Amazon Logs.

S3

Se desideri i log di Amazon S3:

Registri S3

Selezionare S3 logs (Log S3).

Bucket

Scegli il nome del bucket S3 per i tuoi log.

Prefisso del percorso

Inserisci il prefisso per i tuoi log.

Disattiva la crittografia dei log S3

Seleziona se non desideri che i registri S3 siano crittografati.

Modifica delle impostazioni di un progetto di compilazione (AWS CLI)

Per informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta. [Guida di riferimento alla riga di comando](#)

Per aggiornare un CodeBuild progetto con AWS CLI, create un file JSON con le proprietà aggiornate e passate il file al [update-project](#) comando. Tutte le proprietà non contenute nel file di aggiornamento rimangono invariate.

Nel file JSON di aggiornamento, sono richieste solo la name proprietà e le proprietà modificate. La name proprietà identifica il progetto da modificare. Per tutte le strutture modificate, devono essere inclusi anche i parametri richiesti per tali strutture. Ad esempio, per modificare l'ambiente del progetto, sono necessarie environment/computeType le proprietà environment/type and. Ecco un esempio che aggiorna l'immagine dell'ambiente:

```
{
  "name": "<project-name>",
```

```
"environment": {  
  "type": "LINUX_CONTAINER",  
  "computeType": "BUILD_GENERAL1_SMALL",  
  "image": "aws/codebuild/amazonlinux-x86_64-standard:4.0"  
}
```

Se avete bisogno di ottenere i valori correnti delle proprietà per un progetto, usate il [batch-get-projects](#) comando per ottenere le proprietà correnti del progetto che state modificando e scrivete l'output in un file.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

Il *project-info.json* file contiene una serie di progetti, quindi non può essere utilizzato direttamente per aggiornare un progetto. Tuttavia, potete copiare le proprietà che desiderate modificare dal *project-info.json* file e incollarle nel file di aggiornamento come base per le proprietà che desiderate modificare. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un progetto di compilazione \(AWS CLI\)](#).

Modificate il file JSON di aggiornamento come descritto in [Creazione di un progetto di compilazione \(AWS CLI\)](#) e salvate i risultati. Quando hai finito di modificare il file JSON di aggiornamento, esegui il [update-project](#) comando, passando il file JSON di aggiornamento.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

In caso di successo, nell'output viene visualizzato il progetto JSON aggiornato. Se mancano dei parametri obbligatori, nell'output viene visualizzato un messaggio di errore che identifica i parametri mancanti. Ad esempio, questo è il messaggio di errore visualizzato se il `environment/type` parametro è mancante:

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

```
Parameter validation failed:  
Missing required parameter in environment: "type"
```

Modifica delle impostazioni di un progetto di compilazione (AWS SDKs)

Per informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, vedere [AWS SDKs e riferimento agli strumenti](#).

Token di accesso multiplo in CodeBuild

CodeBuild supporta l'approvvigionamento di token di accesso a fornitori terzi dai tuoi segreti all'interno Gestione dei segreti AWS o tramite connessioni. AWS CodeConnections Puoi impostare il tuo segreto o la tua connessione come credenziale predefinita per le interazioni con un provider di terze parti specificato come GitHub Enterprise o GitHub Bitbucket.

Puoi impostare le tue credenziali di origine su tre diversi livelli:

1. Credenziali a livello di account per tutti i progetti: si tratta di credenziali predefinite per tutti i progetti di un account. AWS Verranno utilizzate su un progetto quando non vengono specificate credenziali a livello di progetto o di origine.
2. Credenziali a livello di origine per un repository specifico: si tratta di quando viene definito un segreto o una CodeConnections connessione di Secrets Manager su un'origine del progetto. Queste credenziali verranno utilizzate solo per le operazioni sull'archivio di origine specificato. Ciò consente di configurare più token di accesso con diversi ambiti di autorizzazione nello stesso progetto e di non utilizzare le credenziali predefinite a livello di account.
3. Credenziali di fallback a livello di progetto: è possibile impostare una credenziale di fallback a livello di progetto utilizzandola NO_SOURCE come tipo di fonte principale e definire un segreto o una connessione su di essa. È possibile utilizzarla quando si dispone di più fonti su un progetto, ma si desidera utilizzare le stesse credenziali per esse o quando non si desidera utilizzare le credenziali predefinite a livello di account per il progetto.

Argomenti

- [Passaggio 1: Creare un segreto o una CodeConnections connessione di Secrets Manager](#)
- [Fase 2: concedere al ruolo IAM CodeBuild del progetto l'accesso ai segreti di Secrets Manager](#)
- [Fase 3: Configurare Secrets Manager o i CodeConnections token](#)
- [Opzioni di configurazione aggiuntive](#)

Passaggio 1: Creare un segreto o una CodeConnections connessione di Secrets Manager

Usa le seguenti istruzioni per creare un segreto o una CodeConnections connessione di Secrets Manager:

- [Crea e archivia un token in un segreto di Secrets Manager.](#)
- [Crea una connessione a GitHub](#)
- [Creare una connessione a GitHub Enterprise Server](#)
- [Crea una connessione a Bitbucket](#)

Fase 2: concedere al ruolo IAM CodeBuild del progetto l'accesso ai segreti di Secrets Manager

Note

Prima di continuare, devi avere accesso al token creato in Secrets Manager o CodeConnections.

Per concedere l'accesso al ruolo IAM del CodeBuild progetto a Secrets Manager oppure CodeConnections, è necessario aggiungere la seguente politica IAM.

Per concedere l'accesso al ruolo IAM del CodeBuild progetto

1. Crea un ruolo IAM per il tuo CodeBuild progetto seguendo le istruzioni [Consenti CodeBuild di interagire con altri servizi AWS](#) relative al tuo CodeBuild progetto.
2. Esegui una delle seguenti operazioni:
 - Aggiungi la seguente policy IAM al tuo ruolo di CodeBuild progetto per concedere l'accesso al tuo segreto.

JSON

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



```
    ]
  }
]
}
```

(Facoltativo) Se utilizzi chiavi gestite AWS KMS dal cliente per crittografare un segreto di Secrets Manager, puoi aggiungere la seguente dichiarazione di policy per concedere l'accesso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:iam::*:role/Service*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:SecretARN":
            "arn:aws:iam::*:role/Service*"
        }
      }
    }
  ]
}
```

- Aggiungi la seguente policy IAM al tuo ruolo di CodeBuild progetto per concedere l'accesso alla tua connessione.

JSON

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

```
        "Action": [
            "codeconnections:GetConnectionToken",
            "codeconnections:GetConnection"
        ],
        "Resource": [
            "arn:aws:iam::*:role/Service*"
        ]
    }
}
```

Fase 3: Configurare Secrets Manager o i CodeConnections token

Puoi impostare le tue credenziali di origine su tre diversi livelli con Secrets Manager o CodeConnections token.

Configura Secrets Manager o i CodeConnections token come credenziali a livello di account

È possibile configurare un segreto o una CodeConnections connessione di Secrets Manager come credenziale a livello di account e utilizzarla in un progetto.

Console di gestione AWS

Per configurare una connessione come credenziale a livello di account in Console di gestione AWS

1. Come provider di origine, scegli Bitbucket o Enterprise GitHub. GitHub
2. Per Credential, esegui una delle seguenti operazioni:
 - Scegli Credenziale di origine predefinita per utilizzare la credenziale di origine predefinita del tuo account da applicare a tutti i progetti.
 - a. Se non sei connesso al tuo provider di origine, scegli Gestisci credenziali di origine predefinite.
 - b. Per Tipo di credenziale, scegli un tipo di credenziale.
 - c. Se lo desideri CodeConnections, scegli di utilizzare una connessione esistente o crearne una nuova.

Se hai scelto un tipo di credenziale diverso, per Servizio scegli il servizio che desideri utilizzare per archiviare il token e procedi come segue:

- Se hai scelto di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova e scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
- Se hai scelto di utilizzarlo CodeBuild, inserisci il token o il nome utente e la password dell'app e scegli Salva.
- Scegli Credenziale di origine personalizzata per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni predefinite del tuo account.
 - a. Per Tipo di credenziale, scegli un tipo di credenziale.
 - b. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.

AWS CLI

Per configurare una connessione come credenziale a livello di account in AWS CLI

- Apri un terminale (Linux, macOS o Unix) o prompt dei comandi (Windows). Utilizzare il AWS CLI per eseguire il `import-source-credentials` comando.

Usa il seguente comando per configurare un segreto di Secrets Manager:

```
aws codebuild import-source-credentials \  
  --token "<secret-arn>" \  
  --server-type <source-provider> \  
  --auth-type SECRETS_MANAGER \  
  --region <aws-region>
```

Utilizzate il seguente comando per configurare una CodeConnections connessione:

```
aws codebuild import-source-credentials \  
  --token "<connection-arn>" \  
  --server-type <source-provider> \  
  --auth-type CODECONNECTIONS \  
  --region <aws-region>
```

Questo comando consente di importare un token come credenziali di origine predefinite a livello di account. Quando importi una credenziale utilizzando l'[ImportSourceCredentials](#) API, CodeBuild utilizzerà il token per tutte le interazioni con il provider di origine, come webhook, build status reporting e operazioni git clone, a meno che nel progetto non sia stato configurato un set di credenziali più specifico.

Ora puoi usare il token nel tuo progetto di build ed eseguirlo. Per ulteriori informazioni, consultare [Crea un progetto di compilazione in AWS CodeBuild](#) e [Esegui AWS CodeBuild le build manualmente](#).

Configura più token come credenziali a livello di origine

Per utilizzare i segreti o CodeConnections le connessioni di Secrets Manager come credenziali a livello di origine, fai riferimento direttamente al token nel CodeBuild progetto e avvia una build.

Console di gestione AWS

Per configurare più token come credenziali a livello di origine in Console di gestione AWS

1. Per Provider di origine, scegli. GitHub
2. Per Credential, effettuate una delle seguenti operazioni:
 - Scegli Credenziale di origine predefinita per utilizzare la credenziale di origine predefinita del tuo account da applicare a tutti i progetti.
 - a. Se non sei connesso a GitHub, scegli Gestisci credenziali di origine predefinite.
 - b. Per il tipo di credenziale, scegli GitHub App.
 - c. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.
 - Scegli Credenziale di origine personalizzata per utilizzare una credenziale di origine personalizzata per sostituire le impostazioni predefinite del tuo account.
 - a. Per Tipo di credenziale, scegli App. GitHub
 - b. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.
3. Scegli Aggiungi fonte e ripeti la procedura di scelta del provider di origine e delle credenziali.

AWS CLI

Per configurare più token come credenziali a livello di origine in AWS CLI

- Apri un terminale (Linux, macOS o Unix) o prompt dei comandi (Windows). Utilizzare il AWS CLI per eseguire il comando. create-project

Utilizza il seguente comando:

```
aws codebuild create-project --region <aws-region> \
  --name <project-name> \
  --artifacts type=NO_ARTIFACTS \
  --environment "type=LINUX_CONTAINER,
                  computeType=BUILD_GENERAL1_SMALL,
                  image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
  --service-role <service-role-name> \
  --source "type=GITHUB,
            location=<github-repository-1>,
            auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-1>}" \
  --secondary-sources "type=GITHUB,
                       location=<github-repository-2>,
                       auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-2>},
                       sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project-name>
```

Imposta un fallback delle credenziali di origine a livello di progetto

Per impostare il fallback delle credenziali di origine a livello di progetto, utilizzalo NO_SOURCE come fonte principale del progetto e fai riferimento al token.

```
aws codebuild create-project \
  --name <project-name> \
  --service-role <service-role-name> \
  --artifacts type=NO_ARTIFACTS \
  --environment "type=LINUX_CONTAINER,
                  computeType=BUILD_GENERAL1_SMALL,
                  image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
  --service-role <service-role-name> \
  --source "type=NO_SOURCE,
            auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn>},
```

```

        buildspec=<buildspec>"
    --secondary-sources "type=GITHUB,
                        location=<github-repository>,
                        sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project_name>

```

Quando si utilizza `NO_SOURCE`, una `buildspec` viene in genere fornita all'interno del modello sorgente in quanto non è configurata direttamente per utilizzare una fonte esterna per recuperare la `buildspec`.

Comunemente, una `NO_SOURCE` fonte gestirà la clonazione di tutti i repository pertinenti dall'interno di `buildspec`. Per garantire che le credenziali configurate siano disponibili per tali operazioni, puoi abilitare l'opzione nel `buildspec`. `git-credential-helper`

```

env:
  git-credential-helper: yes

```

Durante la compilazione, CodeBuild leggerà quindi il `AuthServer` campo dal token configurato e utilizzerà le credenziali del token per tutte le richieste git a quel particolare fornitore di sorgenti terzo.

Opzioni di configurazione aggiuntive

È possibile configurare le credenziali a livello di account di Secrets Manager utilizzando CloudFormation modelli. È possibile utilizzare il seguente CloudFormation modello per impostare una credenziale a livello di account:

```

Parameters:
  GitHubToken:
    Type: String
    NoEcho: true
    Default: placeholder
Resources:
  CodeBuildAuthTokenSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token
      Name: codebuild-auth-token
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
            - !Ref GitHubToken

```

```

    - '""}'
  Tags:
    - Key: codebuild:source:provider
      Value: github
    - Key: codebuild:source:type
      Value: personal_access_token
  CodeBuildSecretsManagerAccountCredential:
    Type: AWS::CodeBuild::SourceCredential
  Properties:
    ServerType: GITHUB
    AuthType: SECRETS_MANAGER
    Token: !Ref CodeBuildAuthTokenSecret

```

Note

Se stai anche creando un progetto nello stesso stack, usa l' `CloudFormation DependsOn` attributo per assicurarti che `AccountCredential` venga creato prima del progetto.

È inoltre possibile configurare credenziali multiple a livello di origine di Secrets Manager utilizzando CloudFormation modelli. È possibile utilizzare il seguente CloudFormation modello per utilizzare più token per accedere a più fonti:

```

Parameters:
  GitHubTokenOne:
    Type: String
    NoEcho: true
    Default: placeholder
  GitHubTokenTwo:
    Type: String
    NoEcho: true
    Default: placeholder

Resources:
  CodeBuildSecretsManagerProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: codebuild-multitoken-example
      ServiceRole: <service-role>
      Environment:
        Type: LINUX_CONTAINER

```

```

    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/amazonlinux-x86_64-standard:5.0
  Source:
    Type: GITHUB
    Location: <github-repository-one>
    Auth:
      Type: SECRETS_MANAGER
      Resource: !Ref CodeBuildAuthTokenSecretOne
  SecondarySources:
    - Type: GITHUB
      Location: <github-repository-two>
      Auth:
        Type: SECRETS_MANAGER
        Resource: !Ref CodeBuildAuthTokenSecretTwo
      SourceIdentifier: secondary
  Artifacts:
    Type: NO_ARTIFACTS
  LogsConfig:
    CloudWatchLogs:
      Status: ENABLED
  CodeBuildProjectIAMRoleSecretAccess:
    Type: AWS::IAM::RolePolicy
    Properties:
      RoleName: <role-name>
      PolicyName: CodeBuildProjectIAMRoleSecretAccessPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - secretsmanager:GetSecretValue
            Resource:
              - !Ref CodeBuildAuthTokenSecretOne
              - !Ref CodeBuildAuthTokenSecretTwo
  CodeBuildAuthTokenSecretOne:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token one
      Name: codebuild-auth-token-one
      SecretString:
        !Join
        - ''
        - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
          - !Ref GitHubTokenOne

```



```

    - '{}'
  Tags:
    - Key: codebuild:source:provider
      Value: github
    - Key: codebuild:source:type
      Value: personal_access_token
CodeBuildAuthTokenSecretTwo:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: CodeBuild auth token two
    Name: codebuild-auth-token-two
    SecretString:
      !Join
      - ''
      - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":""'
        - !Ref GitHubTokenTwo
        - '{}'
  Tags:
    - Key: codebuild:source:provider
      Value: github
    - Key: codebuild:source:type
      Value: personal_access_token

```

Eliminare progetti di compilazione in AWS CodeBuild

È possibile utilizzare la CodeBuild console o AWS SDKs eliminare un progetto di compilazione in CodeBuild. AWS CLI Se si elimina un progetto, le sue compilazioni non vengono rimosse.

Warning

Non è possibile eliminare un progetto con una policy delle risorse e compilazioni. Per eliminare un progetto con policy delle risorse e compilazioni, occorre innanzitutto rimuovere la policy o eliminare le compilazioni.

Argomenti

- [Eliminazione di un progetto di compilazione \(console\)](#)
- [Eliminazione di un progetto di compilazione \(AWS CLI\)](#)
- [Eliminazione di un progetto di compilazione \(AWS SDKs\)](#)

Eliminazione di un progetto di compilazione (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Esegui una di queste operazioni:
 - Scegliere il pulsante di opzione accanto al progetto di compilazione da eliminare e quindi scegliere Delete (Elimina).
 - Selezionare il collegamento del progetto di compilazione da eliminare, quindi selezionare Delete (Elimina).

Note

Per impostazione predefinita, vengono visualizzati solo i 10 progetti di compilazione più recenti. Per visualizzare altri progetti di compilazione, selezionare un valore diverso per Projects per page (Progetti per pagina) o utilizzare le frecce avanti e indietro per la visualizzazione dei progetti.

Eliminazione di un progetto di compilazione (AWS CLI)

1. Eseguire il comando `delete-project`:

```
aws codebuild delete-project --name name
```

Sostituire il seguente segnaposto:

- *name*: stringa richiesta. Il nome del progetto di compilazione da eliminare. Per ottenere un elenco di progetti di compilazione disponibili, eseguire il comando `list-projects`. Per ulteriori informazioni, consulta [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#).
2. Se riuscito, non appaiono dati né errori nell'output.

Per ulteriori informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

Eliminazione di un progetto di compilazione (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, vedere [AWS SDKs e riferimento agli strumenti](#).

Get public build project URLs

AWS CodeBuild consente di rendere disponibili al pubblico i risultati della compilazione, i log e gli artefatti dei progetti di compilazione. Ciò consente ai collaboratori dei tuoi repository di sorgenti di visualizzare i risultati e scaricare gli artefatti di una build, senza richiedere loro di avere accesso a un account. AWS

Quando rendi le build del tuo progetto disponibili al pubblico, tutti i risultati, i log e gli artefatti della compilazione di un progetto, incluse le build eseguite quando il progetto era privato, vengono resi disponibili al pubblico. Allo stesso modo, quando rendi privato un progetto di build pubblico, i risultati della compilazione di quel progetto non sono più disponibili al pubblico.

Per informazioni su come modificare la visibilità pubblica dei risultati di costruzione del progetto, consulta [Abilita l'accesso pubblico alla build](#).

CodeBuild fornisce un URL per le build pubbliche del progetto che è unico per il progetto.

Per ottenere l'URL pubblico del progetto di compilazione, utilizzate la procedura seguente.

Per ottenere l'URL di un progetto di compilazione pubblico

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Scegli il link per il progetto di build per il quale desideri ottenere l'URL pubblico.
4. L'URL pubblico viene visualizzato nel campo URL pubblico del progetto nella sezione Configurazione. Puoi scegliere il link per aprire l'URL o copiare l'URL con il pulsante di copia.

Warning

Quando rendi pubblici i risultati di compilazione del progetto, tieni presente quanto segue:

- Tutti i risultati di compilazione, i log e gli artefatti di un progetto, incluse le build eseguite quando il progetto era privato, sono disponibili al pubblico.

- Tutti i log e gli artefatti di compilazione sono disponibili al pubblico. Le variabili di ambiente, il codice sorgente e altre informazioni sensibili potrebbero essere state inserite nei log e negli artefatti di compilazione. È necessario prestare attenzione alle informazioni che vengono emesse nei log di compilazione. Alcune best practice sono:
 - Non memorizzate valori sensibili, in particolare le chiavi di AWS accesso IDs e le chiavi di accesso segrete, nelle variabili di ambiente. Ti consigliamo di utilizzare un Amazon EC2 Systems Manager Parameter Store o Gestione dei segreti AWS di memorizzare valori sensibili.
 - Limita [Procedure consigliate per l'utilizzo dei webhook](#) le entità che possono attivare una build e non memorizzare le specifiche di build nel progetto stesso, per assicurarti che i tuoi webhook siano il più sicuri possibile.
 - Un utente malintenzionato può utilizzare build pubbliche per distribuire artefatti dannosi. Consigliamo agli amministratori del progetto di esaminare tutte le pull request per verificare che la pull request sia una modifica legittima. Ti consigliamo inoltre di convalidare tutti gli artefatti con i relativi checksum per assicurarti che vengano scaricati gli artefatti corretti.

Condividi progetti di costruzione

La condivisione dei progetti consente ai proprietari dei progetti di condividere i propri AWS CodeBuild progetti con altri AWS account o utenti. In questo modello, l'account titolare (proprietario) condivide il proprio progetto con altri account (consumatori). Un consumatore non può modificare o eseguire un progetto.

Argomenti

- [Condividere un progetto](#)
- [Servizi correlati](#)
- [Accedi ai CodeBuild progetti condivisi con te](#)
- [Annullare la condivisione di un progetto condiviso](#)
- [Identifica un progetto condiviso](#)
- [Autorizzazioni per i progetti condivisi](#)

Condividere un progetto

Il consumatore può utilizzare sia la AWS CLI console che la AWS CodeBuild console per visualizzare il progetto e le build che hai condiviso. Il consumatore, inoltre, non può modificare o eseguire il progetto.

Si può creare una condivisione di risorse apposita nella [console AWS RAM](#) o aggiungere il progetto a una condivisione preesistente.

Note

Non è possibile eliminare un progetto con compilazioni, se è stato aggiunto a una condivisione di risorse.

Per condividere un progetto con unità organizzative o un'intera organizzazione, è necessario abilitare la condivisione con AWS Organizations. Per ulteriori informazioni, consulta [Abilitare la condivisione con AWS Organizations](#) nella Guida per l'utente di AWS RAM .

Puoi usare la AWS CodeBuild console, la AWS RAM console o la AWS CLI per condividere un progetto di tua proprietà.

Prerequisiti per la condivisione dei progetti

Prima di iniziare a condividere un progetto, assicurati che il tuo AWS account ne sia proprietario. Un consumatore non può condividere un progetto a cui ha avuto accesso.

Per condividere un progetto di tua proprietà (CodeBuild console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).

Note

Per impostazione predefinita, vengono visualizzati solo i 10 progetti di compilazione più recenti. Per visualizzare altri progetti di compilazione, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

3. Selezionare il progetto da condividere, poi fare clic su Share (Condividi). Per ulteriori informazioni, consulta l'argomento relativo alla [creazione di una condivisione di risorse](#) nella Guida per l'utente di AWS RAM .

Per condividere un progetto di tua proprietà (console)AWS RAM

Consulta [Creating a resource share](#) in Guida per l'utente di AWS RAM .

Per condividere un progetto di cui sei proprietario (AWS RAM comando)

Utilizza il comando [create-resource-share](#).

Come condividere un progetto che si possiede (comando CodeBuild)

Utilizza il comando [put-resource-policy](#):

1. Creare un file denominato `policy.json` e copiarvi quanto segue.

JSON

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

2. Aggiornare `policy.json` con l'ARN del progetto e gli identificatori da coinvolgere nella condivisione. L'esempio seguente concede l'accesso in sola lettura all'utente root per l' AWS account identificato da 123456789012.

JSON

```
{
  "Version": "2012-10-17",
```

```
"Statement": [{
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "123456789012"
    ]
  },
  "Action": [
    "codebuild:BatchGetProjects",
    "codebuild:BatchGetBuilds",
    "codebuild:ListBuildsForProject"],
  "Resource": "arn:aws:codebuild:us-west-2:123456789012:project/my-project"
}]
}
```

3. Esegui il comando [put-resource-policy](#).

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Ottieni l'ARN di condivisione AWS RAM delle risorse.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Ciò restituirà una risposta simile a questa:

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
      "lastUpdatedTime": "<last-update-time>"
    }
  ]
}
```

Dalla risposta, copia il **<resource-share-arn>** valore da utilizzare nel passaggio successivo.

5. Esegui il comando AWS RAM [promote-resource-share-created-from-policy](#).

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-share-arn>
```

Servizi correlati

La condivisione del progetto si integra con AWS Resource Access Manager (AWS RAM), un servizio che consente di condividere AWS le risorse con qualsiasi AWS account o tramite AWS Organizations. Con AWS RAM, creando una condivisione delle risorse, è possibile specificare le risorse e i consumatori interessati. I consumatori possono appartenere a singoli AWS account AWS Organizations, a unità organizzative o a AWS Organizations un'intera organizzazione.

Per ulteriori informazioni, consulta la Guida per l'utente [AWS RAM](#).

Accedi ai CodeBuild progetti condivisi con te

Per accedere a un progetto condiviso, il ruolo IAM di un consumatore richiede l'BatchGetProjects autorizzazione. Puoi allegare la seguente policy al loro ruolo IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetProjects"
  ]
}
```

Per ulteriori informazioni, consulta [Utilizzo di politiche basate sull'identità per AWS CodeBuild](#).

Annullare la condivisione di un progetto condiviso

Annullandone la condivisione, al progetto e alle sue compilazioni può accedere solo il proprietario. Se annulli la condivisione di un progetto, qualsiasi AWS account o utente con cui lo hai condiviso in precedenza non può accedere al progetto o alle relative build.

Per annullare, basta rimuovere il progetto di proprietà dalla relativa condivisione di risorse. Puoi usare la AWS CodeBuild console, la AWS RAM console o AWS CLI per farlo.

Per annullare la condivisione di un progetto condiviso di cui sei proprietario (AWS RAM console)

Consulta [Aggiornamento di una condivisione di risorse](#) nella Guida per l'utente di AWS RAM .

Come annullare la condivisione di un progetto che si possiede (AWS CLI)

Utilizza il comando [disassociate-resource-share](#).

Per annullare la condivisione di un progetto di tua proprietà (comando) CodeBuild

Esegui il [delete-resource-policy](#) comando e specifica l'ARN del progetto che desideri annullare la condivisione:

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Identifica un progetto condiviso

Proprietari e consumatori possono utilizzarli AWS CLI per identificare progetti condivisi.

Per identificare i progetti condivisi con il tuo AWS account o utente (AWS CLI)

Usa il [list-shared-projects](#) comando per restituire i progetti condivisi con te.

Autorizzazioni per i progetti condivisi

Autorizzazioni per i proprietari

Un proprietario di progetto può modificare il progetto e utilizzarlo per eseguire compilazioni.

Autorizzazioni per gli utenti

Un consumatore di progetto può visualizzare il progetto con le sue compilazioni, ma non può modificarlo o avvalersene per eseguire compilazioni.

Etichetta: costruisci progetti

Un tag è un'etichetta di attributo personalizzata che l'utente o AWS assegna a una AWS risorsa. Ogni AWS tag è composto da due parti:

- Una chiave di tag (ad esempio, CostCenter, Environment, Project o Secret). Le chiavi dei tag distinguono tra maiuscole e minuscole
- Un campo facoltativo noto come valore del tag (ad esempio, 111122223333, Production o un nome di team). Non specificare il valore del tag equivale a utilizzare una stringa vuota. Come le chiavi tag, i valori dei tag fanno distinzione tra maiuscole e minuscole.

Tutti questi sono noti come coppie chiave-valore. Per informazioni sul numero di tag che è possibile avere su un progetto e sulle restrizioni sulle chiavi e sui valori dei tag, vedere [Tag](#).

I tag ti aiutano a identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a un CodeBuild progetto che assegni a un bucket S3. [Per ulteriori informazioni sull'utilizzo dei tag, consulta Tagging Best Practices](#).

In CodeBuild, le risorse principali sono il progetto e il gruppo di report. È possibile utilizzare la CodeBuild console AWS CLI CodeBuild APIs, o AWS SDKs per aggiungere, gestire e rimuovere tag per un progetto. Oltre a identificare, organizzare e tracciare il progetto con i tag, puoi utilizzare i tag nelle policy IAM per controllare chi può visualizzare e interagire con il tuo progetto. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Important

Quando si utilizza la funzionalità di capacità riservata, i dati memorizzati nella cache delle istanze del parco istanze, inclusi i file sorgente, i layer Docker e le directory memorizzate nella cache specificate nelle specifiche di costruzione, possono essere accessibili ad altri progetti all'interno dello stesso account. Questa funzionalità è preimpostata e consente ai progetti all'interno dello stesso account di condividere le istanze del parco istanze.

Argomenti

- [Aggiunta di un tag a un progetto](#)
- [Visualizzazione dei tag di un progetto](#)
- [Modifica dei tag di un progetto](#)
- [Rimozione di un tag da un progetto](#)

Aggiunta di un tag a un progetto

L'aggiunta di tag a un progetto può aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. In primo luogo, è possibile aggiungere uno o più tag (coppie chiave-valore) a un progetto. Non dimenticare che sono previsti dei limiti sul numero di tag in un progetto. Esistono limitazioni per i caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Tag](#). Dopo aver creato i tag, puoi creare policy IAM per gestire l'accesso al progetto in base a questi tag. Puoi usare la CodeBuild console o AWS CLI aggiungere tag a un progetto.

Important

Quando si utilizza la funzionalità di capacità riservata, i dati memorizzati nella cache delle istanze della flotta, inclusi i file di origine, i layer Docker e le directory memorizzate nella cache specificate nelle specifiche di costruzione, possono essere accessibili ad altri progetti all'interno dello stesso account. Questa funzionalità è preimpostata e consente ai progetti all'interno dello stesso account di condividere le istanze del parco istanze.

Per ulteriori informazioni sull'aggiunta di tag a un repository al momento della creazione, consulta [Aggiunta di un tag a un progetto \(console\)](#).

Important

Prima di aggiungere un tag a un progetto, assicurati di esaminare tutte le politiche IAM che potrebbero utilizzare i tag per controllare l'accesso a risorse come la creazione di progetti. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Argomenti

- [Aggiunta di un tag a un progetto \(console\)](#)
- [Aggiunta di un tag a un progetto \(AWS CLI\)](#)

Aggiunta di un tag a un progetto (console)

Puoi usare la CodeBuild console per aggiungere uno o più tag a un CodeBuild progetto.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Crea progetti, scegli il nome del progetto in cui desideri aggiungere tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Crea tag progetto.
4. Se non sono stati aggiunti tag al progetto, scegliere Aggiungi tag. In caso contrario, scegliere Edit (Modifica), quindi scegliere Add tag (Aggiungi tag).
5. Per Key (Chiave), inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).
6. (Facoltativo) Per aggiungere un altro tag, scegliere Add tag (Aggiungi tag) .
7. Una volta completata l'aggiunta di tag, scegliere Submit (Inoltra).

Aggiunta di un tag a un progetto (AWS CLI)

Per aggiungere un tag a un progetto al momento della creazione, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#). In `create-project.json`, aggiungi i tuoi tag.

In queste fasi, si assume che sia già installata una versione recente della AWS CLI o che sia aggiornata alla versione corrente. Per ulteriori informazioni, consultare [Installing the AWS Command Line Interface](#).

In caso di successo, questo comando non restituisce alcun risultato.

Visualizzazione dei tag di un progetto

I tag possono aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. Per ulteriori informazioni sull'utilizzo dei tag, consulta il whitepaper [Tagging best practices](#) (Best practice relative al tagging). Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Visualizzazione dei tag di un progetto (console)

Puoi usare la CodeBuild console per visualizzare i tag associati a un CodeBuild progetto.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Crea progetti, scegli il nome del progetto in cui desideri visualizzare i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Crea tag progetto.

Visualizzazione dei tag di un progetto (AWS CLI)

Per visualizzare i tag per un progetto di compilazione, eseguire il comando seguente. Utilizzare il nome del progetto per il parametro `--names`.

```
aws codebuild batch-get-projects --names your-project-name
```

In caso di esito positivo, questo comando restituisce informazioni in formato JSON sul progetto di compilazione che include qualcosa di simile al seguente:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "JanesProject"
  }
}
```

Se il progetto non ha tag, la sezione `tags` è vuota:

```
"tags": []
```

Modifica dei tag di un progetto

È possibile modificare il valore di un tag associato a un progetto. Puoi anche cambiare il nome della chiave, il che equivale a rimuovere il tag corrente e aggiungerne uno diverso con il nuovo nome e lo stesso valore dell'altra chiave. Tieni presente che esistono limitazioni sul numero di caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Tag](#).

Important

La modifica dei tag per un progetto può influire sull'accesso a tale progetto. Prima di modificare il nome (chiave) o il valore di un tag per un progetto, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come la creazione di progetti. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Modifica di un tag di un progetto (Console)

Puoi usare la CodeBuild console per modificare i tag associati a un CodeBuild progetto.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Crea progetti, scegli il nome del progetto in cui desideri modificare i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Crea tag progetto.
4. Scegli Modifica.
5. Esegui una delle seguenti operazioni:
 - Per modificare il tag, immettere un nuovo nome in Key (Chiave). La modifica del nome del tag equivale alla sostituzione del tag esistente con un nuovo tag che ha un nuovo nome chiave.
 - Per modificare il valore di un tag, immettere un nuovo valore. Se si desidera modificare il valore in un valore vuoto, eliminare il valore corrente e lasciare il campo vuoto.
6. Al termine delle modifiche dei tag, scegliere Submit (Inoltra).

Modifica dei tag di un progetto (AWS CLI)

Per aggiungere, modificare o eliminare tag da un progetto di compilazione, vedere [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#). Aggiornare la sezione tags dei dati in formato JSON utilizzati per aggiornare il progetto.

Rimozione di un tag da un progetto

È possibile rimuovere uno o più tag associati a un progetto. La rimozione di un tag non elimina il tag dalle altre AWS risorse ad esso associate.

Important

La rimozione di tag per un progetto può influire sull'accesso a tale progetto. Prima di rimuovere un tag da un progetto, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come la creazione di progetti. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Rimozione di un tag da un progetto (console)

Puoi usare la CodeBuild console per rimuovere l'associazione tra un tag e un CodeBuild progetto.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Crea progetti, scegli il nome del progetto in cui desideri rimuovere i tag.
3. Nel pannello di navigazione scegli Impostazioni. Scegliere Crea tag progetto.
4. Scegli Modifica.
5. Cercare il tag da rimuovere, quindi scegliere Remove tag (Rimuovi tag).
6. Una volta terminata la rimozione di tag, scegliere Submit (Inoltra).

Rimozione di un tag da un progetto (AWS CLI)

Per eliminare uno o più tag da un progetto di compilazione, vedere [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#). Aggiornare la sezione `tags` nei dati formattati JSON con un elenco aggiornato di tag che non contiene quelli che si desidera eliminare. Se si desidera eliminare tutti i tag, aggiornare la sezione `tags` per:

```
"tags: []"
```

Note

Se elimini un progetto di CodeBuild compilazione, tutte le associazioni di tag vengono rimosse dal progetto di build eliminato. Non è necessario rimuovere i tag prima di eliminare un progetto di compilazione.

Usa i runner con AWS CodeBuild

AWS CodeBuild supporta l'integrazione con i runner GitHub Actions, i runner autogestiti e il GitLab runner Buildkite.

Argomenti

- [GitHub I runner Actions ospitati autonomamente in AWS CodeBuild](#)
- [Runner GitLab autogestiti in AWS CodeBuild](#)
- [Buildkite runner autogestito in AWS CodeBuild](#)

GitHub I runner Actions ospitati autonomamente in AWS CodeBuild

Puoi configurare il tuo progetto per configurare gli GitHub Actions runner ospitati autonomamente in CodeBuild contenitori per elaborare i GitHub job del flusso di lavoro Actions. Questo può essere fatto configurando un webhook utilizzando il CodeBuild progetto e aggiornando il flusso di lavoro GitHub Actions YAML per utilizzare runner ospitati autonomamente ospitati su macchine. CodeBuild

I passaggi di alto livello per configurare un CodeBuild progetto per l'esecuzione GitHub dei job Actions sono i seguenti:

1. Se non l'hai già fatto, crea un token di accesso personale o connettiti a un' OAuth app a cui connettere il tuo progetto. GitHub
2. Accedi alla CodeBuild console e crea un CodeBuild progetto con un webhook e configura i filtri webhook.
3. Aggiorna il tuo flusso di lavoro GitHub Actions in YAML GitHub per configurare il tuo ambiente di compilazione.

Per una procedura più dettagliata, consulta. [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#)

Questa funzionalità consente ai job del flusso di lavoro GitHub Actions di ottenere l'integrazione nativa con AWS, il che offre sicurezza e praticità attraverso funzionalità come IAM AWS CloudTrail, Gestione dei segreti AWS integrazione e Amazon VPC. Puoi accedere ai tipi di istanze più recenti, incluse le istanze basate su ARM.

Argomenti

- [Informazioni su CodeBuild GitHub -hosted Actions runner](#)
- [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#)
- [Risolvi i problemi relativi al webhook](#)
- [Le sostituzioni delle etichette sono supportate con il runner -hosted Actions CodeBuild GitHub](#)
- [Immagini di calcolo supportate con il runner -hosted Actions CodeBuild GitHub](#)

Informazioni su CodeBuild GitHub -hosted Actions runner

Di seguito sono riportate alcune domande comuni sul runner CodeBuild -hosted Actions GitHub .

Quando devo includere l'immagine e le sostituzioni delle istanze nell'etichetta?

Puoi includere le sostituzioni di immagini e istanze nell'etichetta per specificare un ambiente di compilazione diverso per ciascuno dei processi del flusso di lavoro Actions. GitHub Questa operazione può essere eseguita senza la necessità di creare più CodeBuild progetti o webhook. Ad esempio, ciò è utile quando è necessario utilizzare una [matrice per i lavori del flusso di lavoro](#).

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      image:${{ matrix.os }}
      instance-size:${{ matrix.size }}
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

Note

Le virgolette potrebbero essere necessarie se `runs-on` sono presenti più etichette contenenti il contesto GitHub Actions.

Posso usare CloudFormation per questa funzionalità?

Sì, puoi includere un gruppo di filtri nel tuo CloudFormation modello che specifichi un filtro GitHub Actions workflow Job Event nel webhook del progetto.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Per ulteriori informazioni, consulta [Filtra gli eventi GitHub webhook \(\)CloudFormation](#).

Se hai bisogno di aiuto per configurare le credenziali del progetto nel tuo CloudFormation modello, consulta [AWS::CodeBuild::SourceCredential](#) la Guida per l'AWS CloudFormation utente per ulteriori informazioni.

Come posso mascherare i segreti quando utilizzo questa funzione?

Per impostazione predefinita, i segreti stampati nel registro non sono mascherati. Se desideri mascherare i tuoi segreti, puoi usare la seguente sintassi: `::add-mask::` *value* Quello che segue è un esempio di come puoi usare questa sintassi nel tuo YAML:

```
name: Secret Job
on: [push]
jobs:
  Secret-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    env:
      SECRET_NAME: "secret-name"
    steps:
      - run: echo "::add-mask::$SECRET_NAME"
```

Per ulteriori informazioni, vedere [Mascherare un valore in un accesso](#). GitHub

Posso ricevere eventi webhook di GitHub Actions da più repository all'interno di un singolo progetto?

CodeBuild supporta webhook di livello organizzativo e globale, che ricevono eventi da un'organizzazione o un'impresa specifica. Per ulteriori informazioni, consulta [GitHub webhook globali e organizzativi](#).

Quali aree geografiche supportano l'utilizzo di un Actions CodeBuild runner ospitato GitHub ?

CodeBuild runner -hosted GitHub Actions sono supportati in tutte le regioni. CodeBuild Per ulteriori informazioni su Regioni AWS dove CodeBuild è disponibile, consulta [AWS Servizi per regione](#).

Quali piattaforme supportano l'utilizzo di un GitHub Actions runner CodeBuild ospitato?

CodeBuild-hosted GitHub Actions runner sono supportati sia su Amazon [AWS Lambda](#) EC2 che su Compute. Puoi utilizzare le seguenti piattaforme: Amazon Linux 2, Amazon Linux 2023, Ubuntu e Windows Server Core 2019. Per ulteriori informazioni, consultare [EC2 calcola immagini](#) e [Immagini di calcolo Lambda](#).

Tutorial: configura un GitHub Actions CodeBuild runner ospitato

Questo tutorial mostra come configurare i CodeBuild progetti per eseguire i job GitHub Actions. Per ulteriori informazioni sull'utilizzo di GitHub Actions, CodeBuild vedere [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#).

Per completare questo tutorial, è necessario innanzitutto:

- Connettiti con un token di accesso personale, un segreto, un'app o OAuth GitHub un'app di Secrets Manager. Se desideri connetterti con un' OAuth app, devi utilizzare la CodeBuild console per farlo. Se desideri creare un token di accesso personale, puoi utilizzare la CodeBuild console o utilizzare l'[ImportSourceCredentials API](#). Per ulteriori istruzioni, consulta [GitHub e l'accesso a GitHub Enterprise Server in CodeBuild](#).
- Connect CodeBuild al tuo GitHub account. A tale scopo, puoi effettuare una delle seguenti operazioni:
 - Puoi aggiungerlo GitHub come provider di origine nella console. Puoi connetterti con un token di accesso personale, un segreto di Secrets Manager, OAuth un'app o GitHub un'app. Per istruzioni, consulta [GitHub e l'accesso a GitHub Enterprise Server in CodeBuild](#).
 - Puoi importare GitHub le tue credenziali tramite l'[ImportSourceCredentials API](#). Questo può essere fatto solo con un token di accesso personale. Se ti connetti tramite un' OAuth app, devi invece connetterti utilizzando la console. Per istruzioni, consulta [Connect GitHub con un token di accesso \(CLI\)](#).

Note

Questa operazione deve essere eseguita solo se non ti sei connesso GitHub al tuo account.

Fase 1: Creare un CodeBuild progetto con un webhook

In questo passaggio, creerai un CodeBuild progetto con un webhook e lo esaminerai nella GitHub console. Puoi anche scegliere GitHub Enterprise come fornitore di origine. Per ulteriori informazioni sulla creazione di un webhook in GitHub Enterprise, consulta [GitHub webhook manuali](#).

Per creare un CodeBuild progetto con un webhook

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
3. In Tipo di progetto, scegli Runner project.

In Runner:

- a. Per il provider Runner, scegli. GitHub
- b. Per la posizione del Runner, scegli Repository.
- c. Per Repository URL in Repository, scegli repository-name. <https://github.com/user-name/>

 Note

Per impostazione predefinita, il progetto riceverà solo `WORKFLOW_JOB_QUEUED` eventi per un singolo repository. Se desideri ricevere eventi per tutti gli archivi all'interno di un'organizzazione o di un'azienda, consulta. [GitHub webhook globali e organizzativi](#)

4. • In Environment (Ambiente):
 - Scegli un'immagine Environment supportata e Calcola. Tieni presente che hai la possibilità di sovrascrivere le impostazioni dell'immagine e dell'istanza utilizzando un'etichetta nel flusso di lavoro GitHub Actions (YAML). Per ulteriori informazioni, consulta [Passaggio 2: aggiorna il flusso di lavoro di GitHub Actions \(YAML\)](#)
 - In Buildspec:
 - Nota che il tuo buildspec verrà ignorato a meno che non venga aggiunto come etichetta. `buildspec-override:true` Invece, lo CodeBuild sovrascriverà per utilizzare i comandi che configureranno il runner self-hosted.
5. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.
6. Apri la GitHub console all'indirizzo <https://github.com/user-name/repository-name/settings/hooks> per verificare che un webhook sia stato creato e che sia abilitato a fornire eventi Workflow jobs.

Passaggio 2: aggiorna il flusso di lavoro di GitHub Actions (YAML)

In questo passaggio, aggiornerai il file YAML del flusso di lavoro GitHub Actions [GitHub](#) per configurare l'ambiente di compilazione e utilizzare i runner self-hosted di GitHub Actions in.

CodeBuild Per ulteriori informazioni, consulta [Uso delle etichette con runner ospitati autonomamente](#) e [Le sostituzioni delle etichette sono supportate con il runner -hosted Actions CodeBuild GitHub](#)

Aggiorna il flusso di lavoro GitHub Actions (YAML).

Vai [GitHub](#) e aggiorna l'[runs-on](#) impostazione nel tuo flusso di lavoro GitHub Actions YAML per configurare il tuo ambiente di compilazione. A tale scopo, puoi effettuare una delle seguenti operazioni:

- Puoi specificare il nome del progetto e l'ID di esecuzione, nel qual caso la build utilizzerà la configurazione del progetto esistente per il calcolo, l'immagine, la versione dell'immagine e la dimensione dell'istanza. Il nome del progetto è necessario per collegare le impostazioni AWS relative del lavoro GitHub Actions a un progetto specifico CodeBuild. Includendo il nome del progetto in YAML, CodeBuild è consentito richiamare lavori con le impostazioni di progetto corrette. Fornendo l'ID di esecuzione, CodeBuild mapperà la build a specifiche esecuzioni del flusso di lavoro e interromperà la compilazione quando l'esecuzione del flusso di lavoro viene annullata. Per ulteriori informazioni, consulta [githubcontext](#).

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

Note

Assicurati che *<project-name>* corrisponda al nome del progetto che hai creato nel passaggio precedente. Se non corrisponde, non CodeBuild elaborerà il webhook e il flusso di lavoro GitHub Actions potrebbe bloccarsi.

Di seguito è riportato un esempio di flusso di lavoro GitHub Actions YAML:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    steps:
      - run: echo "Hello World!"
```

- Puoi anche sovrascrivere l'immagine e il tipo di calcolo nell'etichetta. Vedi [Immagini di calcolo supportate con il runner -hosted Actions CodeBuild GitHub](#) per un elenco di immagini curate. Per l'utilizzo di immagini personalizzate, consulta [Le sostituzioni delle etichette sono supportate con il runner -hosted Actions CodeBuild GitHub](#). Il tipo di calcolo e l'immagine nell'etichetta sovrascriveranno le impostazioni di ambiente del progetto. Per sovrascrivere le impostazioni dell'ambiente per una build di calcolo CodeBuild EC2 o Lambda, usa la seguente sintassi:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    image:<environment-type>-<image-identifier>
    instance-size:<instance-size>
```

Di seguito è riportato un esempio di flusso di lavoro YAML di Actions: GitHub

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:arm-3.0
        instance-size:small
    steps:
      - run: echo "Hello World!"
```

- Puoi sovrascrivere la flotta utilizzata per la tua build nell'etichetta. Ciò sovrascriverà le impostazioni del parco veicoli configurate nel progetto per utilizzare il parco veicoli specificato. Per ulteriori informazioni, consulta [Run si basa su flotte a capacità riservata](#). Per sovrascrivere le impostazioni del tuo parco macchine per una build di calcolo Amazon EC2, usa la seguente sintassi:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
```

Per sovrascrivere sia la flotta che l'immagine utilizzate per la build, usa la seguente sintassi:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
    image:<environment-type>-<image-identifier>
```

Di seguito è riportato un esempio di flusso di lavoro YAML di GitHub Actions:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      fleet:myFleet
      image:arm-3.0
    steps:
      - run: echo "Hello World!"
```

- Per eseguire i lavori GitHub Actions su un'immagine personalizzata, puoi configurare un'immagine personalizzata nel tuo CodeBuild progetto ed evitare di fornire un'etichetta di sostituzione dell'immagine. CodeBuild utilizzerà l'immagine configurata nel progetto se non viene fornita alcuna etichetta di sostituzione dell'immagine.
- Facoltativamente, puoi fornire etichette diverse da quelle supportate CodeBuild . Queste etichette verranno ignorate allo scopo di sovrascrivere gli attributi della build, ma non falliranno la richiesta del webhook. Ad esempio, l'aggiunta `testLabel` come etichetta non impedirà l'esecuzione della build.

Note

Se una dipendenza fornita da GitHub -hosted runners non è disponibile nell'ambiente CodeBuild, puoi installare la dipendenza utilizzando GitHub Actions nell'esecuzione del flusso di lavoro. Ad esempio, puoi usare l'[setup-python](#) azione per installare Python per il tuo ambiente di compilazione.

Esegui i comandi `buildspec` nelle fasi `INSTALL`, `PRE_BUILD` e `POST_BUILD`

Per impostazione predefinita, CodeBuild ignora qualsiasi comando `buildspec` durante l'esecuzione di una build Actions ospitata autonomamente. GitHub Per eseguire i comandi `buildspec` durante la compilazione, è `buildspec-override:true` possibile aggiungere come suffisso all'etichetta:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

```
buildspec-override:true
```

Utilizzando questo comando, CodeBuild creerà una cartella chiamata `actions-runner` nella cartella di origine principale del contenitore. Quando il runner GitHub Actions si avvia durante la BUILD fase, il runner verrà eseguito nella `actions-runner` directory.

Esistono diverse limitazioni quando si utilizza un override buildspec in una build di Actions ospitata autonomamente: GitHub

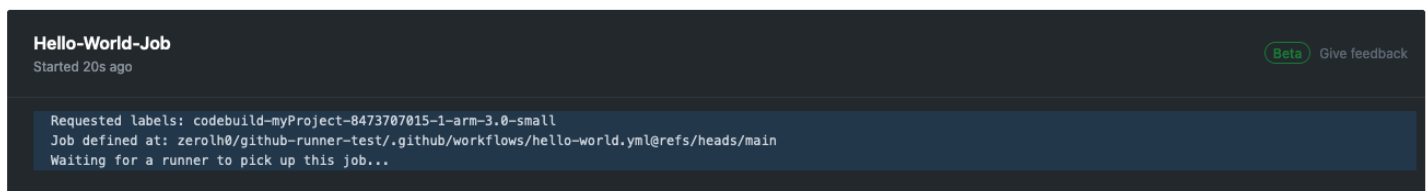
- CodeBuild non eseguirà i comandi buildspec durante la BUILD fase, poiché il runner ospitato autonomamente viene eseguito durante la fase. BUILD
- CodeBuild non scaricherà alcuna fonte primaria o secondaria durante la fase. DOWNLOAD_SOURCE
Se hai configurato un file buildspec, solo quel file verrà scaricato dalla fonte principale del progetto.
- Se un comando build fallisce nella INSTALL fase PRE_BUILD or, non CodeBuild avvierà il runner self-hosted e il processo di workflow GitHub Actions dovrà essere annullato manualmente.
- CodeBuild recupera il token runner durante la DOWNLOAD_SOURCE fase, che ha una scadenza di un'ora. Se le tue INSTALL fasi PRE_BUILD o superano un'ora, il runner token potrebbe scadere prima dell'inizio del runner ospitato autonomamente. GitHub

Passaggio 3: rivedi i risultati

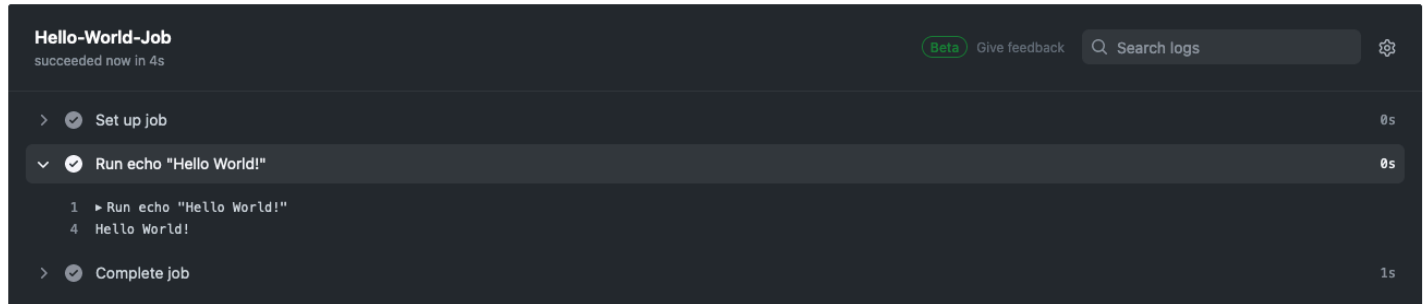
Ogni volta che viene eseguito un flusso di lavoro GitHub Actions, riceve CodeBuild gli eventi del processo del flusso di lavoro tramite il webhook. Per ogni job del workflow, CodeBuild avvia una build per eseguire un Actions runner temporaneo. GitHub Il runner è responsabile dell'esecuzione di un singolo processo di workflow. Una volta completato il processo, il runner e il processo di compilazione associato verranno immediatamente interrotti.

Per visualizzare i registri dei lavori del flusso di lavoro, accedi al tuo repository in GitHub, scegli Azioni, scegli il flusso di lavoro desiderato, quindi scegli il Job specifico per il quale desideri esaminare i log.

Puoi rivedere le etichette richieste nel registro mentre il lavoro è in attesa di essere ritirato da un runner che si occupa di hosting autonomo. CodeBuild



Una volta completato il lavoro, potrai visualizzarne il registro.



GitHub Opzioni di configurazione di Actions Runner

È possibile specificare le seguenti variabili di ambiente nella configurazione del progetto per modificare la configurazione di installazione dei corridori ospitati autonomamente.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

CodeBuild registrerà i corridori ospitati autonomamente con il nome dell'organizzazione specificato come valore di questa variabile di ambiente. Per ulteriori informazioni sulla registrazione dei corridori a livello di organizzazione e sulle autorizzazioni necessarie, consulta [Creare una configurazione per un corridore per un' just-in-time](#)organizzazione.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

CodeBuild registrerà i corridori ospitati autonomamente con il nome aziendale specificato come valore di questa variabile di ambiente. Per ulteriori informazioni sulla registrazione dei runner a livello aziendale e sulle autorizzazioni necessarie, consulta [Create configuration for a runner for an Enterprise. just-in-time](#)

Note

Per impostazione predefinita, gli Enterprise runner non sono disponibili negli archivi delle organizzazioni. Affinché i runner ospitati autonomamente possano iniziare a lavorare nel flusso di lavoro, potrebbe essere necessario configurare le impostazioni di accesso del gruppo di runner. Per ulteriori informazioni, consulta [Rendere disponibili i runner aziendali](#) nei repository.

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

CodeBuild registrerà i corridori ospitati autonomamente nell'ID intero del gruppo runner memorizzato come valore di questa variabile di ambiente. Per impostazione predefinita, questo valore è 1. Per ulteriori informazioni sui gruppi di corridori ospitati autonomamente, consulta [Gestione dell'accesso ai corridori ospitati autonomamente](#) tramite gruppi.

CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME

Per configurare la registrazione dei corridori a livello di organizzazione utilizzando il file YAML del flusso di lavoro GitHub Actions, puoi utilizzare la seguente sintassi:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        organization-registration-name:myOrganization
    steps:
      - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME

Per configurare la registrazione dei runner di livello aziendale utilizzando il file YAML del flusso di lavoro GitHub Actions, è possibile utilizzare la seguente sintassi:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        enterprise-registration-name:myEnterprise
    steps:
      - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

Per configurare la registrazione dei corridori a uno specifico ID di gruppo di corridori utilizzando il file YAML del flusso di lavoro GitHub Actions, puoi utilizzare la seguente sintassi:

```

name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      registration-group-id:3
    steps:
      - run: echo "Hello World!"

```

Filter Actions webhook events () GitHub CloudFormation

La seguente parte di un CloudFormation modello in formato YAML crea un gruppo di filtri che attiva una build quando restituisce true. Il seguente gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions con un nome di flusso di lavoro corrispondente all'espressione regolare.

\[CI-CodeBuild\]

```

CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      FilterGroups:
        - - Type: EVENT
          Pattern: WORKFLOW_JOB_QUEUED
        - Type: WORKFLOW_NAME
          Pattern: \[CI-CodeBuild\]

```

Filter GitHub Actions webhook events ()AWS CDK

Il AWS CDK modello seguente crea un gruppo di filtri che attiva una build quando restituisce true. Il seguente gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions.

```
import { aws_codebuild as codebuild } from 'aws-cdk-lib';
import { EventAction, FilterGroup } from "aws-cdk-lib/aws-codebuild";

const source = codebuild.Source.gitHub({
  owner: 'owner',
  repo: 'repo',
  webhook: true,
  webhookFilters: [FilterGroup.inEventOf(EventAction.WORKFLOW_JOB_QUEUED)],
})
```

Eventi webhook di Filter GitHub Actions (Terraform)

Il seguente modello Terraform crea un gruppo di filtri che attiva una build quando restituisce true. Il seguente gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions.

```
resource "aws_codebuild_webhook" "example" {
  project_name = aws_codebuild_project.example.name
  build_type   = "BUILD"
  filter_group {
    filter {
      type      = "EVENT"
      pattern = "WORKFLOW_JOB_QUEUED"
    }
  }
}
```

Filter GitHub Actions webhook events ()AWS CLI

AWS CLI I comandi seguenti creano un progetto GitHub Actions runner ospitato autonomamente con un gruppo di filtri per le richieste di lavoro del flusso di lavoro di GitHub Actions che attiva una build quando restituisce true.

```
aws codebuild create-project \
--name <project name> \
--source "{\"type\":\"GITHUB\",\"location\":\"<repository location>\",\"buildspec\":\"\\\"\\\"}\" \
```

```
--artifacts {"\"type\": \"NO_ARTIFACTS\""} \
--environment {"\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/amazonlinux-x86_64-standard:5.0\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\""} \
--service-role "<service role ARN>"
```

```
aws codebuild create-webhook \
--project-name <project name> \
--filter-groups "[[{"\"type\": \"EVENT\", \"pattern\": \"WORKFLOW_JOB_QUEUED\"}]]"
```

Risolvi i problemi relativi al webhook

Problema: il webhook configurato [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#) non funziona o il processo del flusso di lavoro è in sospeso. GitHub

Possibili cause:

- Il tuo evento webhook Workflow jobs potrebbe non riuscire ad attivare una build. Esamina i registri delle risposte per visualizzare la risposta o il messaggio di errore.
- I tuoi lavori vengono assegnati al runner agent errato a causa della configurazione delle etichette. Questo problema può verificarsi quando uno dei lavori all'interno di una singola esecuzione del flusso di lavoro ha meno etichette rispetto a un altro lavoro. Ad esempio, se hai due lavori con le seguenti etichette nello stesso flusso di lavoro, esegui:
 - Job 1: codebuild-myProject-\${{ github.run_id }}-\${{ github.run_attempt }}
 - Job 2:codebuild-myProject-\${{ github.run_id }}-\${{ github.run_attempt }}, instance-size:medium

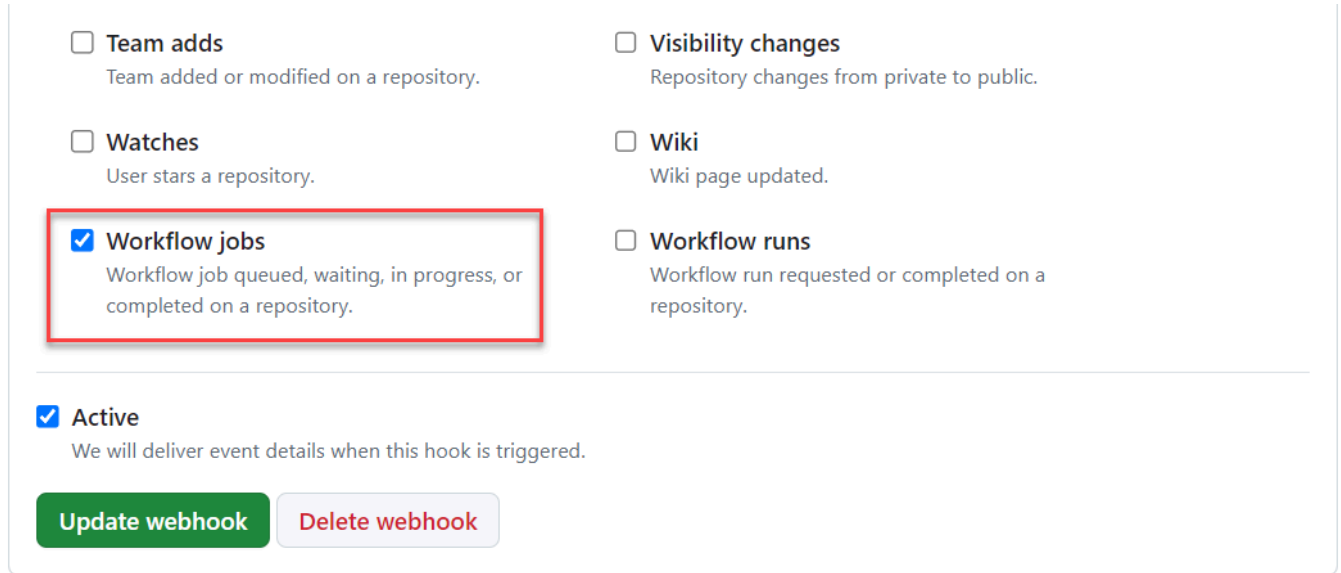
Quando si esegue il routing di un job GitHub Actions ospitato autonomamente, lo GitHub indirizzerà a qualsiasi runner con tutte le etichette specificate per il lavoro. Questo comportamento significa che Job 1 può essere prelevato dal runner creato per Job 1 o Job 2, ma Job 2 può essere prelevato solo dal runner creato per Job 2 poiché ha un'etichetta aggiuntiva. Se Job 1 viene scelto dal runner creato per Job 2, Job 2 si bloccherà poiché il Job 1 runner non ha l'instance-size:mediumetichetta.

Soluzioni consigliate:

Quando crei più lavori all'interno della stessa esecuzione del flusso di lavoro, utilizza lo stesso numero di sostituzioni di etichette per ogni lavoro o assegna a ciascun lavoro un'etichetta personalizzata, ad esempio o. job1 job2

Se l'errore persiste, usa le seguenti istruzioni per eseguire il debug del problema.

1. Apri la GitHub console all'indirizzo `https://github.com/user-name/repository-name/settings/hooks` per visualizzare le impostazioni del webhook del tuo repository. In questa pagina, vedrai un webhook creato per il tuo repository.
2. Scegli Modifica e conferma che il webhook è abilitato a fornire eventi di lavoro Workflow.



☐ Team adds
Team added or modified on a repository.

☐ Watches
User stars a repository.

☒ Workflow jobs
Workflow job queued, waiting, in progress, or completed on a repository.

☐ Visibility changes
Repository changes from private to public.

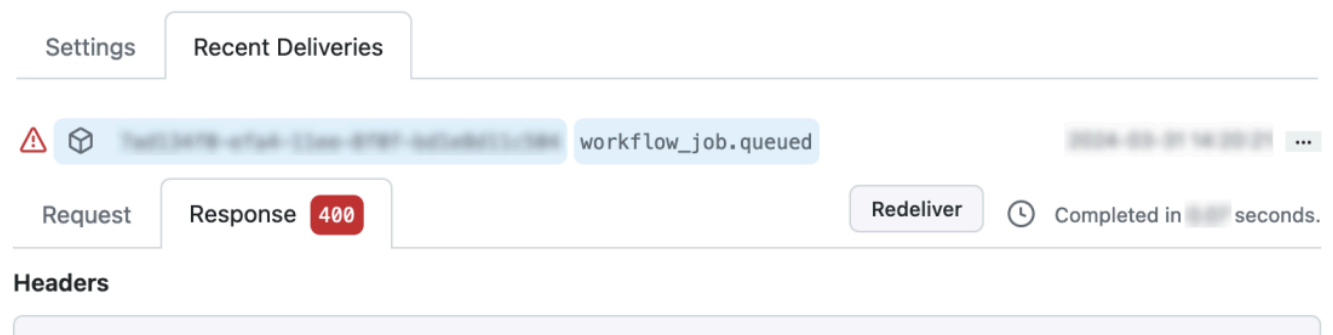
☐ Wiki
Wiki page updated.

☐ Workflow runs
Workflow run requested or completed on a repository.

☒ Active
We will deliver event details when this hook is triggered.

Update webhook Delete webhook

3. Vai alla scheda Consegne recenti, trova l'`workflow_job.queued` evento corrispondente ed espandi l'evento.
4. Controlla il campo delle etichette nel Payload e assicurati che sia come previsto.
5. Infine, esamina la scheda Risposta, poiché contiene la risposta o il messaggio di errore restituito da CodeBuild.



Settings Recent Deliveries

⚠️ workflow_job.queued

Request Response 400 Redeliver Completed in seconds.

Headers

6. In alternativa, è possibile eseguire il debug degli errori dei webhook utilizzando's. GitHub APIs. Puoi visualizzare le consegne recenti per un webhook utilizzando l'API [List deliveries](#) for a repository webhook:

```
gh api \
```

```
-H "Accept: application/vnd.github+json" \  
-H "X-GitHub-API-Version: 2022-11-28" \  
/repos/owner/repo/hooks/hook-id/deliveries
```

Dopo aver trovato la distribuzione del webhook di cui stai cercando di eseguire il debug e aver annotato l'ID di consegna, puoi utilizzare l'API webhook [Get a delivery for a repository](#). CodeBuild la risposta del payload di consegna del webhook è disponibile nella sezione: **response**

```
gh api \  
-H "Accept: application/vnd.github+json" \  
-H "X-GitHub-API-Version: 2022-11-28" \  
/repos/owner/repo/hooks/hook-id/deliveries/delivery-id
```

Problema: i trigger GitHub Actions with [deployment protection](#) rules abilitati vengono compilati CodeBuild prima dell'approvazione della distribuzione.

Possibili cause: CodeBuild recupera la distribuzione e l'ambiente associati al job GitHub Actions, se esistono, per verificare se sono approvati. Se CodeBuild non riesce a recuperare la distribuzione o l'ambiente, la CodeBuild build potrebbe essere attivata prematuramente.

Soluzioni consigliate: verifica che le credenziali associate ai CodeBuild progetti dispongano delle autorizzazioni di lettura per le distribuzioni e le azioni interne. GitHub

Le sostituzioni delle etichette sono supportate con il runner -hosted Actions CodeBuild GitHub

Nel tuo flusso di lavoro GitHub Actions (YAML), puoi fornire una serie di sostituzioni di etichette che modificano la build del tuo runner ospitato autonomamente. Tutte le build non riconosciute da CodeBuild verranno ignorate ma non falliranno la richiesta del webhook. Ad esempio, il seguente flusso di lavoro YAML include sostituzioni per immagine, dimensione dell'istanza, flotta e buildspec:

```
name: Hello World  
on: [push]  
jobs:  
  Hello-World-Job:  
    runs-on:  
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}  
      image:${{ matrix.os }}
```

```
instance-size:${{ matrix.size }}
fleet:myFleet
buildspec-override:true
strategy:
  matrix:
    include:
      - os: arm-3.0
        size: small
      - os: linux-5.0
        size: large
steps:
  - run: echo "Hello World!"
```

Note

Se il processo del flusso di lavoro è in sospeso GitHub, consulta [Risolvi i problemi relativi al webhook](#) e [Utilizzo di etichette personalizzate per indirizzare i lavori](#).

codebuild-*<project-name>*-\${{github.run_id}}-
\${{github.run_attempt}}(obbligatorio)

- Ad esempio: codebuild-fake-project-\${{ github.run_id }}-
\${{ github.run_attempt }}
- Obbligatorio per tutti i flussi di lavoro GitHub Actions YAMLs. *<project name>* deve essere uguale al nome del progetto per il quale è configurato il webhook runner self-hosted.

image:*<environment-type>*-*<image-identifier>*

- Ad esempio: image:arm-3.0
- Sostituisce l'immagine e il tipo di ambiente utilizzati quando si avvia la build self-hosted runner con un'immagine curata. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner -hosted Actions CodeBuild GitHub](#)
- Per sovrascrivere l'immagine e il tipo di ambiente utilizzati con un'immagine personalizzata, usa image:custom-*<environment-type>*-*<custom-image-identifier>*
- Ad esempio: image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0

 Note

Se l'immagine personalizzata si trova in un registro privato, vedi. [Configura una credenziale di registro privata per i corridori ospitati autonomamente](#)

`instance-size:` *<instance-size>*

- Ad esempio: `instance-size:medium`
- Sostituisce il tipo di istanza utilizzato all'avvio della build self-hosted runner. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner -hosted Actions CodeBuild GitHub](#)

`fleet:` *<fleet-name>*

- Ad esempio: `fleet:myFleet`
- Sostituisce le impostazioni del parco veicoli configurate nel progetto per utilizzare il parco veicoli specificato. Per ulteriori informazioni, consulta [Run si basa su flotte a capacità riservata](#).

`buildspec-override:` *<boolean>*

- Ad esempio: `buildspec-override:true`
- Consente alla build di eseguire i comandi buildspec nelle fasi, e INSTALL PRE_BUILDPOST_BUILD, se impostata su. `true`

Sovrascrittura di un'etichetta singola (legacy)

CodeBuild consente di fornire più sostituzioni in una singola etichetta utilizzando quanto segue:

- Per sovrascrivere le impostazioni dell'ambiente per una build di calcolo Amazon EC2/Lambda, usa la seguente sintassi:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-  
${{ github.run_attempt }}-<environment-type>-<image-identifier>-<instance-size>
```

- Per sovrascrivere le impostazioni del tuo parco macchine per la build di calcolo di Amazon EC2, utilizza la seguente sintassi:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}-
fleet-<fleet-name>
```

- Per sovrascrivere sia la flotta che l'immagine utilizzate per la build, usa la seguente sintassi:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-image-<image-version>-fleet-<fleet-name>
```

- Per eseguire i comandi buildspec durante la compilazione, -with-buildspec possono essere aggiunti come suffisso all'etichetta:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>-with-buildspec
```

- Facoltativamente, puoi fornire un override della dimensione dell'istanza senza sovrascrivere l'immagine. Per le build di Amazon EC2, puoi escludere sia il tipo di ambiente che l'identificatore dell'immagine. Per le build Lambda, puoi escludere l'identificatore dell'immagine.

Immagini di calcolo supportate con il runner -hosted Actions CodeBuild GitHub

Nell'etichetta in cui hai configurato[Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#), puoi sovrascrivere le impostazioni dell'ambiente Amazon EC2 utilizzando i valori nelle prime tre colonne. CodeBuild fornisce le seguenti immagini di calcolo di Amazon EC2. Per ulteriori informazioni sull'

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge gpu_small	Amazon Linux 2023	aws/codebuild/amazonlinux-x	al/standard/5.0

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
		gpu_large		86_64-standard:5.0	
linux-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Nessuno
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nessuno

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nessuno

Inoltre, puoi sovrascrivere le impostazioni dell'ambiente Lambda utilizzando i seguenti valori. Per ulteriori informazioni sul calcolo CodeBuild Lambda, vedere. [Esegui build su calcolo AWS Lambda](#) CodeBuild supporta le seguenti immagini di calcolo Lambda:

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto11	4GB			
		8GB			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza			
	ruby3.2				

Per ulteriori informazioni, consultare [Modi e tipi di calcolo dell'ambiente di creazione](#) e [Immagini Docker fornite da CodeBuild](#).

Runner GitLab autogestiti in AWS CodeBuild

GitLab offre due modalità di esecuzione per eseguire i GitLab lavori nella CI/CD pipeline. Una modalità è GitLab costituita dai runner ospitati, gestiti GitLab e completamente integrati con. GitLab L'altra modalità è costituita dai runner autogestiti, che consentono di creare un ambiente personalizzato per eseguire i job nella pipeline CI/CD. GitLab

I passaggi di alto livello per configurare un CodeBuild progetto per eseguire i job della pipeline GitLab CI/CD sono i seguenti:

1. Se non l'hai già fatto, connettiti con un' OAuth app a cui connettere il tuo progetto. GitLab
2. Accedi alla CodeBuild console e crea un CodeBuild progetto con un webhook e configura i filtri webhook.
3. Aggiorna YAML della tua pipeline GitLab CI/CD per configurare il tuo ambiente di compilazione. GitLab

Per una procedura più dettagliata, consulta. [Tutorial: configura un CodeBuild runner ospitato GitLab](#)

Questa funzionalità consente l'integrazione nativa con i job della pipeline GitLab CI/CD AWS, che offre sicurezza e praticità attraverso funzionalità come IAM AWS CloudTrail e Amazon VPC. Puoi accedere ai tipi di istanze più recenti, incluse le istanze basate su ARM.

Argomenti

- [Informazioni su -hosted CodeBuild runner GitLab](#)
- [Tutorial: configura un CodeBuild runner ospitato GitLab](#)
- [Le sostituzioni delle etichette sono supportate con -hosted runner CodeBuild GitLab](#)
- [Immagini di calcolo supportate con il runner -hosted CodeBuild GitLab](#)

Informazioni su -hosted CodeBuild runner GitLab

Di seguito sono riportate alcune domande comuni su CodeBuild GitLab -hosted runner.

Quali tipi di sorgenti sono supportati per CodeBuild GitLab -hosted runner?

CodeBuild-hosted GitLab runner sono supportati per il tipo di sorgente and. GITLAB
GITLAB_SELF_MANAGED

Quando devo includere l'immagine e le sostituzioni delle istanze nell'etichetta?

È possibile includere le sostituzioni di immagini e istanze nell'etichetta per specificare un ambiente di compilazione diverso per ciascuno dei processi della pipeline CI/CD. GitLab Questo può essere fatto senza la necessità di creare più progetti o webhook. CodeBuild

Posso usare CloudFormation per questa funzionalità?

Sì, puoi includere un gruppo di filtri nel tuo CloudFormation modello che specifichi un filtro per gli eventi GitLab di workflow job nel webhook del tuo progetto.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Per ulteriori informazioni, consulta [Filtra gli GitLab eventi webhook \(\)CloudFormation](#).

Se hai bisogno di aiuto per configurare le credenziali del progetto nel tuo CloudFormation modello, consulta [AWS::CodeBuild::SourceCredential](#) la Guida per l'AWS CloudFormation utente per ulteriori informazioni.

Come posso mascherare i segreti quando utilizzo questa funzione?

Per impostazione predefinita, i segreti stampati nel registro non sono mascherati. Se desideri mascherare i tuoi segreti, puoi farlo aggiornando le impostazioni delle variabili di CI/CD ambiente:

Per mascherare i segreti in GitLab

1. Nelle tue GitLab Impostazioni, scegli CI/CD.
2. In Variabili, scegli Modifica per il segreto che desideri mascherare.

3. In Visibilità, seleziona Variabile maschera, quindi scegli Aggiorna variabile per salvare le modifiche.

Posso ricevere eventi GitLab webhook da più progetti all'interno di un singolo gruppo?

CodeBuild supporta i webhook di gruppo, che ricevono eventi da un gruppo specifico. GitLab Per ulteriori informazioni, consulta [GitLab webhook di gruppo](#).

Posso eseguire un lavoro in docker executor per il runner autogestito? Ad esempio, voglio eseguire un processo di pipeline su un'immagine specifica per mantenere lo stesso ambiente di compilazione in un contenitore separato e isolato.

È possibile eseguire il runner GitLab autogestito CodeBuild con un'immagine specifica [creando il progetto con un'immagine personalizzata o sovrascrivendo l'immagine nel file](#). `.gitlab-ci.yml`

Con quale esecutore viene eseguito il runner autogestito? CodeBuild

Il runner autogestito CodeBuild viene eseguito con l'esecutore di shell, dove la build viene eseguita localmente insieme al GitLab runner in esecuzione all'interno del contenitore docker.

Posso fornire comandi buildspec insieme al runner autogestito?

Sì, è possibile aggiungere comandi buildspec insieme al runner autogestito. Puoi fornire il file `buildspec.yml` nel tuo GitLab repository e utilizzare il tag nella sezione Tag del lavoro. **buildspec-override:true** Per ulteriori informazioni, consulta [Nome del file buildspec e posizione di storage](#).

Quali regioni supportano l'utilizzo di un runner ospitato? CodeBuild GitLab

CodeBuild-hosted GitLab runner sono supportati in tutte le regioni. CodeBuild Per ulteriori informazioni su Regioni AWS dove CodeBuild è disponibile, consulta [AWS Servizi per regione](#).

Quali piattaforme supportano l'utilizzo di un GitLab runner CodeBuild ospitato?

CodeBuild GitLab -i runner ospitati sono supportati sia su Amazon [AWS Lambda](#) EC2 che su Compute. Puoi utilizzare le seguenti piattaforme: Amazon Linux 2, Amazon Linux 2023, Ubuntu e Windows Server Core 2019. Per ulteriori informazioni, consultare [EC2 calcola immagini](#) e [Immagini di calcolo Lambda](#).

Tutorial: configura un CodeBuild runner ospitato GitLab

Questo tutorial mostra come configurare i CodeBuild progetti per eseguire lavori di pipeline GitLab CI/CD. Per ulteriori informazioni sull'utilizzo GitLab o sulla gestione GitLab automatica con CodeBuild, consulta [Runner GitLab autogestiti in AWS CodeBuild](#)

Per completare questo tutorial, è necessario innanzitutto:

- Connect con un' OAuth app utilizzando CodeConnections. Tieni presente che quando ti connetti a un' OAuth app, devi utilizzare la CodeBuild console per farlo. Per ulteriori istruzioni, consulta [GitLab accesso in CodeBuild](#).
- Connect CodeBuild al tuo GitLab account. A tale scopo, puoi aggiungerlo GitLab come provider di origine nella console. Per istruzioni, consulta [GitLab accesso in CodeBuild](#).

Note

Questa operazione deve essere eseguita solo se non ti sei connesso GitLab al tuo account. Con questa funzione, sono CodeBuild necessarie autorizzazioni aggiuntive, ad esempio `create_runner` e `manage_runner` dall' OAuth app. Se esistono CodeConnections per un determinato GitLab account, non richiede automaticamente gli aggiornamenti delle autorizzazioni. A tale scopo, puoi accedere alla CodeConnections console e creare una connessione fittizia allo stesso GitLab account per attivare la riautorizzazione e ottenere le autorizzazioni aggiuntive. In questo modo, tutte le connessioni esistenti possono utilizzare la funzionalità runner. Una volta completata, puoi eliminare la connessione fittizia.

Fase 1: Creare un CodeBuild progetto con un webhook

In questo passaggio, creerai un CodeBuild progetto con un webhook e lo esaminerai nella GitLab console.

Per creare un CodeBuild progetto con un webhook

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).

In Tipo di progetto, scegli Runner project.

- In Runner:
 - Per il provider Runner, scegli. GitLab
 - Per Credential, scegli una delle seguenti opzioni:
 - Scegli Credenziale di origine predefinita. La connessione predefinita applica una GitLab connessione predefinita a tutti i progetti.
 - Scegli Credenziali di origine personalizzate. La connessione personalizzata applica una GitLab connessione personalizzata che sostituisce le impostazioni predefinite dell'account.

 Note

Se non hai già creato una connessione con il tuo provider, dovrai crearne una nuova GitLab . Per istruzioni, consulta [Connect CodeBuild a GitLab](#).

- Per la posizione del Runner, scegli Repository.
 - Per Repository, scegli il nome del tuo progetto in GitLab specificando il percorso del progetto con lo spazio dei nomi.
 - In Environment (Ambiente):
 - Scegli un'immagine Environment supportata e Calcola. Tieni presente che hai la possibilità di sovrascrivere le impostazioni dell'immagine e dell'istanza utilizzando un'etichetta nella pipeline GitLab CI/CD YAML. Per ulteriori informazioni, consulta [Passaggio 2: crea un file.gitlab-ci.yml nel tuo repository](#).
 - In Buildspec:
 - Nota che il tuo buildspec verrà ignorato a meno che non venga aggiunto come etichetta. `buildspec-override:true` Invece, lo CodeBuild sovrascriverà per utilizzare i comandi che configureranno il runner autogestito.
 -
3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.
 4. Apri la GitLab console all'indirizzo <https://gitlab.com/user-name/repository-name/-/hooks> per verificare che un webhook sia stato creato e che sia abilitato a fornire eventi Workflow jobs.

Passaggio 2: crea un file.gitlab-ci.yml nel tuo repository

In questo passaggio, creerai un .gitlab-ci.yml file per configurare l'ambiente di compilazione e utilizzare i runner [GitLab](#) autogestiti in. GitLab CodeBuild Per ulteriori informazioni, consulta [Utilizzare corridori autogestiti](#).

Aggiorna la tua pipeline GitLab CI/CD YAML

Accedi al tuo repository `https://gitlab.com/user-name/project-name/-/tree/branch-name` e crea un .gitlab-ci.yml file. Puoi configurare il tuo ambiente di compilazione effettuando una delle seguenti operazioni:

- Puoi specificare il nome del CodeBuild progetto, nel qual caso la build utilizzerà la configurazione del progetto esistente per il calcolo, l'immagine, la versione dell'immagine e la dimensione dell'istanza. Il nome del progetto è necessario per collegare le impostazioni AWS relative del GitLab lavoro a un progetto specifico CodeBuild. Includendo il nome del progetto in YAML, CodeBuild è consentito richiamare lavori con le impostazioni di progetto corrette.

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

`$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME` è necessario per mappare la compilazione a specifiche esecuzioni di processi di pipeline e interrompere la compilazione quando l'esecuzione della pipeline viene annullata.

Note

Assicurati che *<project-name>* corrisponda al nome del progetto in cui hai creato. CodeBuild Se non corrisponde, non CodeBuild elaborerà il webhook e la pipeline GitLab CI/CD potrebbe bloccarsi.

Di seguito è riportato un esempio di pipeline CI/CD YAML GitLab :

```
workflow:
  name: HelloWorld
  stages:           # List of stages for jobs, and their order of execution
    - build

  build-job:        # This job runs in the build stage, which runs first.
```

```

stage: build
script:
  - echo "Hello World!"
tags:
  - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME

```

- Puoi anche sovrascrivere l'immagine e il tipo di calcolo nel tag. Vedi [Immagini di calcolo supportate con il runner -hosted CodeBuild GitLab](#) per un elenco di immagini curate. Per l'utilizzo di immagini personalizzate, consulta [Le sostituzioni delle etichette sono supportate con -hosted runner CodeBuild GitLab](#). Il tipo di calcolo e l'immagine nel tag sovrascriveranno le impostazioni di ambiente del progetto. Per sovrascrivere le impostazioni dell'ambiente per una build di calcolo Amazon EC2, usa la seguente sintassi:

```

tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - image:<environment-type>-<image-identifier>
  - instance-size:<instance-size>

```

Di seguito è riportato un esempio di pipeline CI/CD YAML GitLab :

```

stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small

```

- Puoi sovrascrivere la flotta utilizzata per la tua build nel tag. Ciò sovrascriverà le impostazioni della flotta configurate nel tuo progetto per utilizzare la flotta specificata. Per ulteriori informazioni, consulta [Run si basa su flotte a capacità riservata](#). Per sovrascrivere le impostazioni del tuo parco macchine per una build di calcolo Amazon EC2, usa la seguente sintassi:

```

tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>

```

Per sovrascrivere sia la flotta che l'immagine utilizzate per la build, usa la seguente sintassi:

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - fleet:<fleet-name>
  - image:<environment-type>-<image-identifier>
```

Di seguito è riportato un esempio di pipeline GitLab CI/CD YAML:

```
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:myFleet
    - image:arm-3.0
```

- Per eseguire i processi della pipeline GitLab CI/CD su un'immagine personalizzata, è possibile configurare un'immagine personalizzata nel CodeBuild progetto ed evitare di fornire un'etichetta di sostituzione dell'immagine. CodeBuild utilizzerà l'immagine configurata nel progetto se non viene fornita alcuna etichetta di sostituzione dell'immagine.

Dopo aver confermato le modifiche a `.gitlab-ci.yml`, verrà attivata una GitLab pipeline e `build-job` invierà una notifica webhook che avvierà la compilazione. CodeBuild

Esegui i comandi `buildspec` nelle fasi `INSTALL`, `PRE_BUILD` e `POST_BUILD`

Per impostazione predefinita, CodeBuild ignora qualsiasi comando `buildspec` quando si esegue una build autogestita. GitLab Per eseguire i comandi `buildspec` durante la compilazione, `buildspec-override:true` possono essere aggiunti come suffisso a: `tags`

```
tags:
  - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  - buildspec-override:true
```

Utilizzando questo comando, CodeBuild creerà una cartella chiamata `gitlab-runner` nella cartella di origine principale del contenitore. Quando il GitLab corridore parte durante la BUILD fase, il corridore verrà eseguito nella `gitlab-runner` directory.

Esistono diverse limitazioni quando si utilizza un override buildspec in una build autogestita: GitLab

- CodeBuild non eseguirà i comandi buildspec durante la BUILD fase, poiché il runner autogestito viene eseguito durante la fase. BUILD
- CodeBuild non scaricherà alcuna fonte primaria o secondaria durante la fase. DOWNLOAD_SOURCE. Se hai configurato un file buildspec, solo quel file verrà scaricato dalla fonte principale del progetto.
- Se un comando build fallisce nella INSTALL fase PRE_BUILD o, non CodeBuild avvierà il runner autogestito e il processo della pipeline GitLab CI/CD dovrà essere annullato manualmente.
- CodeBuild recupera il token runner durante la DOWNLOAD_SOURCE fase, che ha una scadenza di un'ora. Se le tue INSTALL fasi PRE_BUILD o superano un'ora, il runner token potrebbe scadere prima dell'inizio del runner autogestito GitLab .

Passaggio 3: rivedi i risultati

Ogni volta che si tratta di un lavoro in GitLab CI/CD pipeline run occurs, CodeBuild would receive the CI/CD pipeline job events through the webhook. For each job in the CI/CD pipeline, CodeBuild starts a build to run an ephemeral GitLab runner. The runner is responsible for executing a single CI/CD pipeline. Una volta completato il lavoro, il runner e il processo di compilazione associato verranno immediatamente interrotti.

Per visualizzare i log dei lavori della CI/CD pipeline, accedi al tuo repository in GitLab, scegli Build, Jobs, quindi scegli il Job specifico per il quale desideri esaminare i log.

Puoi rivedere le etichette richieste nel registro mentre il lavoro è in attesa di essere completato da un runner autogestito. CodeBuild

Filtra gli GitLab eventi webhook ()CloudFormation

La seguente parte di un CloudFormation modello in formato YAML crea un gruppo di filtri che attiva una build quando restituisce true. Il seguente gruppo di filtri specifica un GitLab CI/CD pipeline job request with a CI/CD nome di pipeline corrispondente all'espressione regolare. `\[CI-CodeBuild\]`

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
```

```
Name: MyProject
ServiceRole: service-role
Artifacts:
  Type: NO_ARTIFACTS
Environment:
  Type: LINUX_CONTAINER
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:5.0
Source:
  Type: GITLAB
  Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
Triggers:
  Webhook: true
  ScopeConfiguration:
    Name: group-name
    Scope: GITLAB_GROUP
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
    - Type: WORKFLOW_NAME
      Pattern: \[CI-CodeBuild\]
```

Le sostituzioni delle etichette sono supportate con -hosted runner CodeBuild GitLab

Nella tua pipeline GitLab CI/CD YAML, puoi fornire una serie di sostituzioni di etichette che modificano la tua build di runner autogestita. Tutte le build non riconosciute da CodeBuild verranno ignorate ma non falliranno la richiesta del webhook. Ad esempio, il seguente YAML include le sostituzioni per immagine, dimensione dell'istanza, flotta e buildspec:

```
workflow:
  name: HelloWorld
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small
    - fleet:myFleet
```

```
- buildspec-override:true
```

codebuild-*<project-name>*-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME(richiesto)

- Ad esempio: codebuild-myProject-\$CI_PROJECT_ID-\$CI_PIPELINE_IID-\$CI_JOB_NAME
- Obbligatorio per tutte le GitLab pipeline CI/CD. YAMLs *<project name>* deve essere uguale al nome del progetto per il quale è configurato il webhook runner autogestito.

image:*<environment-type>*-*<image-identifier>*

- Ad esempio: image:arm-3.0
- Sostituisce l'immagine e il tipo di ambiente utilizzati all'avvio della build del runner autogestito. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner - hosted CodeBuild GitLab](#)
- Per sovrascrivere l'immagine e il tipo di ambiente utilizzati con un'immagine personalizzata, usa image:custom-*<environment-type>*-*<custom-image-identifier>*
- Ad esempio: image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0

 Note

Se l'immagine personalizzata si trova in un registro privato, vedi. [Configura una credenziale di registro privata per i corridori ospitati autonomamente](#)

instance-size:*<instance-size>*

- Ad esempio: instance-size:small
- Sostituisce il tipo di istanza utilizzato all'avvio della build runner autogestita. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner -hosted CodeBuild GitLab](#)

fleet:*<fleet-name>*

- Ad esempio: fleet:myFleet

- Sostituisce le impostazioni del parco veicoli configurate nel progetto per utilizzare il parco veicoli specificato. Per ulteriori informazioni, consulta [Run si basa su flotte a capacità riservata](#).

`buildspec-override:` *<boolean>*

- Ad esempio: `buildspec-override:true`
- Consente alla build di eseguire i comandi `buildspec` nelle fasi, e `INSTALL` `PRE_BUILD` `POST_BUILD`, se impostata su. `true`

Immagini di calcolo supportate con il runner -hosted CodeBuild GitLab

Nell'etichetta in cui hai configurato [Tutorial: configura un CodeBuild runner ospitato GitLab](#), puoi sovrascrivere le impostazioni dell'ambiente Amazon EC2 utilizzando i valori nelle prime tre colonne. CodeBuild fornisce le seguenti immagini di calcolo di Amazon EC2. Per ulteriori informazioni sull'

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine	Definizione
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
		gpu_small			
		gpu_large			
linux-ec2	latest	small	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64	Nessuno
		medium			
		large			

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine	Definizione
				-base:latest	
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nessuno

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine	Definizione
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine	Definizione
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nessuno

Inoltre, puoi sovrascrivere le impostazioni dell'ambiente Lambda utilizzando i seguenti valori. Per ulteriori informazioni sul calcolo CodeBuild Lambda, vedere. [Esegui build su calcolo AWS Lambda](#) CodeBuild supporta le seguenti immagini di calcolo Lambda:

Tipo di ambiente	Versione di runtime	Dimensioni istanza			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto11	4GB			
		8GB			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				

Tipo di ambiente	Versione di runtime	Dimensioni istanza			
	ruby3.2				

Per ulteriori informazioni, consultare [Modi e tipi di calcolo dell'ambiente di creazione](#) e [Immagini Docker fornite da CodeBuild](#).

Buildkite runner autogestito in AWS CodeBuild

Puoi configurare il tuo progetto per configurare i runner Buildkite ospitati autonomamente in CodeBuild contenitori per elaborare i tuoi job Buildkite. Questo può essere fatto configurando un webhook utilizzando il CodeBuild progetto e aggiornando i passaggi YAML della pipeline di Buildkite per utilizzare runner ospitati autonomamente ospitati su macchine. CodeBuild

I passaggi di alto livello per configurare un progetto per eseguire i job di Buildkite sono i seguenti: CodeBuild

- Passa alla CodeBuild console e crea un CodeBuild progetto con la configurazione del tipo Buildkite runner (project runner).
- Aggiungi un `job.scheduled` webhook alla tua organizzazione Buildkite.
- Aggiorna i passaggi YAML della pipeline di Buildkite in Buildkite per configurare il tuo ambiente di compilazione.

Per una procedura più dettagliata, vedi. [Tutorial: configura un CodeBuild Buildkite runner ospitato](#)
Questa funzionalità consente ai lavori di Buildkite di ottenere l'integrazione nativa con AWS, che offre sicurezza e praticità attraverso funzionalità come IAM e Amazon VPC. Gestione dei segreti AWS AWS CloudTrail Puoi accedere ai tipi di istanze più recenti, incluse le istanze basate su ARM.

Informazioni sul Buildkite runner CodeBuild ospitato

Di seguito sono riportate alcune domande comuni sul runner Buildkite -hosted CodeBuild.

Quando devo includere l'immagine e le sostituzioni delle istanze nell'etichetta?

Puoi includere le sostituzioni di immagini e istanze nell'etichetta per specificare un ambiente di compilazione diverso per ciascuno dei tuoi lavori Buildkite. Questo può essere fatto senza

la necessità di creare più CodeBuild progetti o webhook. Ad esempio, questo è utile quando è necessario utilizzare una [matrice per i lavori Buildkite](#).

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "${matrix.os}"
      instance-size: "${matrix.size}"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "a12-5.0"
        size:
          - "small"
          - "large"
```

È possibile CodeBuild creare webhook all'interno di Buildkite automaticamente?

Attualmente, Buildkite richiede che tutti i webhook vengano creati manualmente utilizzando la loro console. Puoi seguire il tutorial su [Tutorial: configura un CodeBuild Buildkite runner ospitato](#) per creare manualmente un webhook Buildkite nella console Buildkite.

Posso usare CloudFormation per creare webhook Buildkite?

CloudFormation attualmente non è supportato per i webhook Buildkite runner, poiché Buildkite richiede che i webhook vengano creati manualmente utilizzando la loro console.

Quali regioni supportano l'utilizzo di un Buildkite runner ospitato? CodeBuild

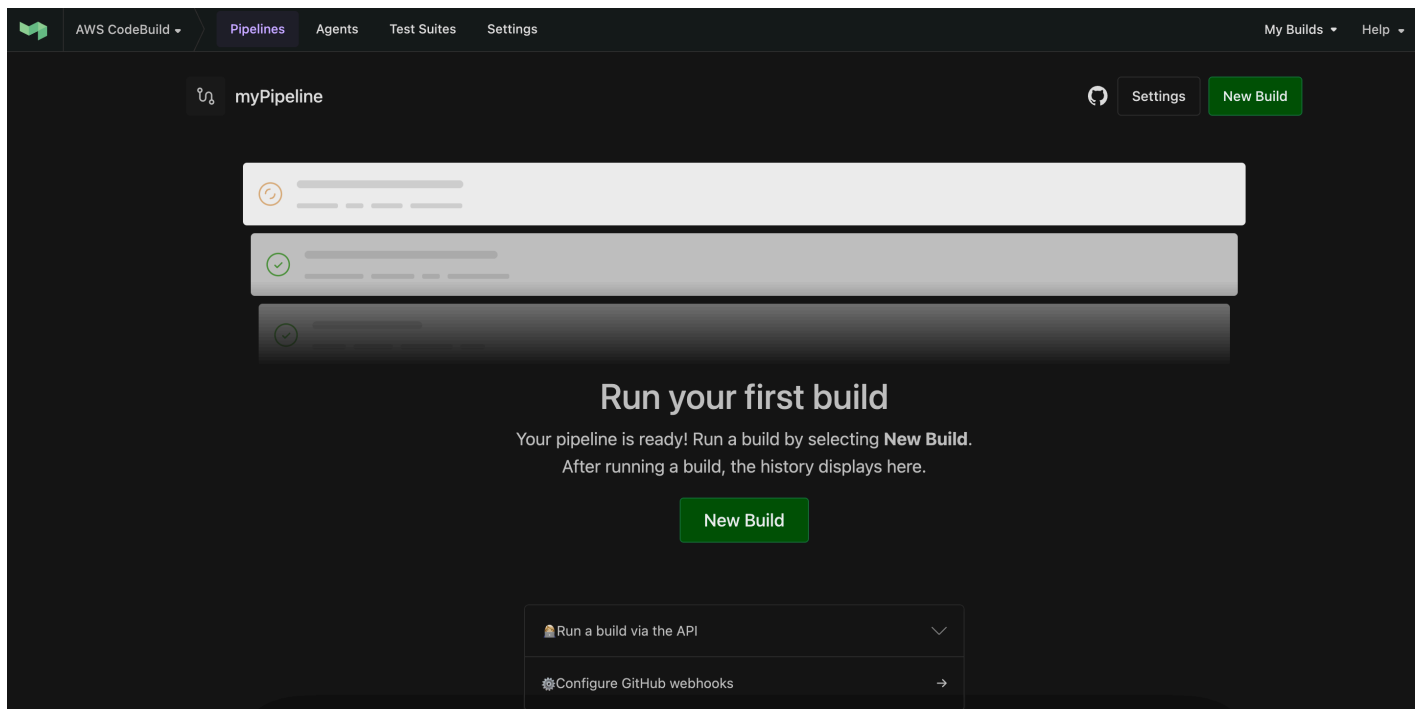
CodeBuild runner Buildkite -hosted sono supportati in tutte le regioni. CodeBuild [Per ulteriori informazioni sulle AWS regioni in cui CodeBuild è disponibile, consulta AWS Servizi per regione.](#)

Tutorial: configura un CodeBuild Buildkite runner ospitato

Questo tutorial mostra come configurare i tuoi CodeBuild progetti per eseguire i job di Buildkite. Per ulteriori informazioni sull'utilizzo di Buildkite, vedere. CodeBuild [Buildkite runner autogestito in AWS CodeBuild](#)

Per completare questo tutorial, è necessario innanzitutto:

- Avere accesso a un'organizzazione Buildkite. [Per ulteriori informazioni sulla configurazione di un account e di un'organizzazione Buildkite, puoi seguire questo tutorial introduttivo.](#)
- Crea una pipeline, un cluster e una coda di Buildkite configurati per utilizzare runner ospitati autonomamente. [Per ulteriori informazioni sulla configurazione di queste risorse, puoi fare riferimento al Buildkite Pipeline Setup Tutorial.](#)



Passaggio 1: generare un token agente Buildkite

In questo passaggio, genererai un token agente all'interno di Buildkite che verrà utilizzato per autenticare i runner ospitati autonomamente. CodeBuild [Per ulteriori informazioni su questa risorsa, consulta Buildkite Agent Tokens.](#)

Per generare un token dell'agente Buildkite

1. Nel tuo cluster Buildkite, scegli Agent Tokens, quindi scegli Nuovo token.
2. Aggiungi una descrizione al token e fai clic su Crea token.
3. Salva il valore del token dell'agente, poiché verrà utilizzato in seguito durante la configurazione CodeBuild del progetto.

Agent Tokens > New

Description — Required

myToken

Describe the set of agents this token is for

Allowed IP Addresses

0.0.0.0 ::/0

Restrict which network addresses are allowed to use this agent token. Use space-separated, IPv4 **CIDR notation**, for example:
192.0.2.0/24 198.51.100.12 25c7:c056:9943:1e01::/64 .

Create Token

Fase 2: Creare un CodeBuild progetto con un webhook

Per creare un CodeBuild progetto con un webhook

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
 2. Crea un progetto di compilazione ospitato autonomamente. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
- Nella configurazione del progetto, seleziona Progetto Runner. In Runner:
 - Per il provider Runner, scegli Buildkite.
 - Per il token dell'agente Buildkite, scegli Crea un nuovo token agente utilizzando la pagina di creazione segreta. Ti verrà richiesto di creare un nuovo token segreto Gestione dei segreti AWS con un valore segreto uguale al token dell'agente Buildkite che hai generato sopra.
 - (Facoltativo) Se desideri utilizzare credenziali CodeBuild gestite per il tuo lavoro, seleziona il provider del repository di origine del tuo lavoro nelle opzioni delle credenziali di origine di Buildkite e verifica che le credenziali siano configurate per il tuo account. Inoltre, verifica che la tua pipeline Buildkite utilizzi Checkout utilizzando HTTPS.

 Note

Buildkite richiede le credenziali di origine all'interno dell'ambiente di compilazione per estrarre il codice sorgente del tuo lavoro. Vedi le opzioni di [Autenticazione di Buildkite su un repository privato](#) credenziali di origine disponibili.

- (Facoltativo) In Environment:

- Scegli un'immagine Environment supportata e Calcola.

Nota che hai la possibilità di sovrascrivere le impostazioni dell'immagine e dell'istanza utilizzando un'etichetta nei passaggi YAML di Buildkite. Per ulteriori informazioni, consulta [Fase 4: Aggiorna i passaggi della pipeline di Buildkite](#).

- (Facoltativo) In Buildspec:

- Il tuo buildspec verrà ignorato per impostazione predefinita a meno che non venga aggiunto come etichetta. `buildspec-override: "true"` Invece, lo CodeBuild sovrascriverà per utilizzare i comandi che configureranno il runner self-hosted.

 Note

CodeBuild non supporta i file buildspec per le build di runner self-hosted di Buildkite. Per le specifiche di compilazione in linea, dovrai abilitarle nel tuo buildspec se hai configurato le credenziali di origine gestite [git-credential-helper](#) CodeBuild

3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.
4. Salva l'URL del payload e i valori segreti dal popup Create Webhook. Segui le istruzioni nel popup per creare un nuovo webhook dell'organizzazione Buildkite o continua con la sezione successiva.

Fase 3: Crea un webhook all'interno di Buildkite CodeBuild

In questo passaggio, utilizzerai i valori Payload URL e Secret del CodeBuild webhook per creare un nuovo webhook all'interno di Buildkite. Questo webhook verrà utilizzato per attivare le build entro CodeBuild l'avvio di un job Buildkite valido.

Per creare un nuovo webhook in Buildkite

1. Vai alla pagina delle impostazioni della tua organizzazione Buildkite.

2. In Integrazioni, seleziona Servizi di notifica.
3. Scegli Aggiungi accanto alla casella Webhook. Nella pagina Aggiungi notifica Webhook, utilizza la seguente configurazione:
 - a. In Webhook URL, aggiungi il valore dell'URL di payload salvato.
 - b. In Token, verifica che l'opzione Invia il token come X-Buildkite-Token sia selezionata. Aggiungi il valore segreto del tuo webhook al campo Token.
 - c. In, verifica che l'opzione Invia il token come X-Buildkite-Token sia selezionata. Aggiungi il valore segreto del tuo webhook al campo Token.
 - d. In Eventi, seleziona l'evento `job.scheduled` webhook.
 - e. (Facoltativo) In Pipelines, puoi facoltativamente scegliere di attivare solo le build per una pipeline specifica.
4. Scegliete Aggiungi notifica Webhook.

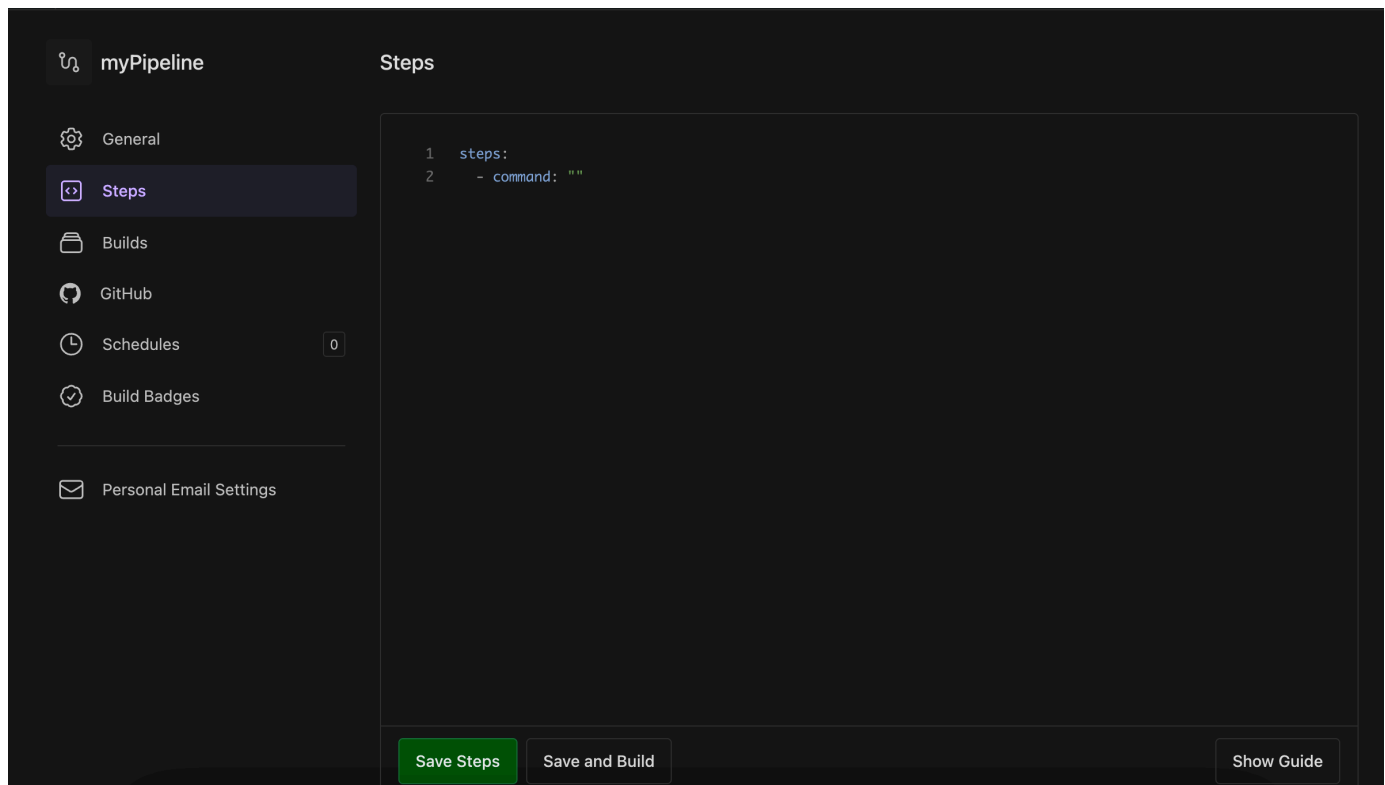
Fase 4: Aggiorna i passaggi della pipeline di Buildkite

In questo passaggio, aggiornerai i passaggi della tua pipeline Buildkite per aggiungere le etichette necessarie e le sostituzioni opzionali. Per l'elenco completo delle sostituzioni di etichette supportate, consulta [Le sostituzioni delle etichette sono supportate con il runner Buildkite ospitato CodeBuild](#)

Aggiorna i passaggi della pipeline

1. Vai alla pagina dei passaggi della pipeline Buildkite selezionando la pipeline Buildkite, scegliendo Impostazioni e quindi scegliendo Steps.

Se non l'hai già fatto, scegli Converti in passaggi YAML.



2. Come minimo, dovrai specificare un [tag dell'agente Buildkite](#) che faccia riferimento al nome della tua pipeline. CodeBuild Il nome del progetto è necessario per collegare le impostazioni AWS relative del job Buildkite a un progetto specifico. CodeBuild Includendo il nome del progetto in YAML, CodeBuild è consentito richiamare lavori con le impostazioni di progetto corrette.

```
agents:
  project: "codebuild-<project name>"
```

Di seguito è riportato un esempio di passaggi della pipeline di Buildkite con solo il tag di etichetta del progetto:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "echo \"Hello World\""
```

Puoi anche sovrascrivere l'immagine e il tipo di calcolo nell'etichetta. [Immagini di calcolo supportate con il runner Buildkite ospitato CodeBuild](#) Per un elenco delle immagini disponibili, consulta la sezione. Il tipo di calcolo e l'immagine nell'etichetta sovrascriveranno le impostazioni

di ambiente del progetto. Per sovrascrivere le impostazioni dell'ambiente per una build di calcolo CodeBuild EC2 o Lambda, usa la seguente sintassi:

```
agents:  
  project: "codebuild-<project name>"  
  image: "<environment-type>-<image-identifier>"  
  instance-size: "<instance-size>"
```

Di seguito è riportato un esempio di passaggi della pipeline di Buildkite con sostituzioni delle dimensioni di immagini e istanze:

```
agents:  
  project: "codebuild-myProject"  
  image: "arm-3.0"  
  instance-size: "small"  
steps:  
  - command: "echo \"Hello World\""
```

Puoi sovrascrivere la flotta utilizzata per la tua build nell'etichetta. Ciò sovrascriverà le impostazioni del parco veicoli configurate nel progetto per utilizzare il parco veicoli specificato. Per ulteriori informazioni, consulta [Esegui build su flotte a capacità riservata](#).

Per sovrascrivere le impostazioni del tuo parco macchine per una build di calcolo Amazon EC2, usa la seguente sintassi:

```
agents:  
  project: "codebuild-<project name>"  
  fleet: "<fleet-name>"
```

Per sovrascrivere sia la flotta che l'immagine utilizzate per la build, usa la seguente sintassi:

```
agents:  
  project: "codebuild-<project name>"  
  fleet: "<fleet-name>"  
  image: "<environment-type>-<image-identifier>"
```

Di seguito è riportato un esempio di passaggi della pipeline di Buildkite con sostituzioni di flotta e immagini:

```
agents:
  project: "codebuild-myProject"
  fleet: "myFleet"
  image: "arm-3.0"
steps:
  - command: "echo \"Hello World\""
```

3. Puoi scegliere di eseguire i comandi buildspec in linea durante la build del runner Buildkite ospitata autonomamente (vedi per maggiori dettagli). [Esegui i comandi buildspec per le fasi INSTALL, PRE_BUILD e POST_BUILD](#) Per specificare che la CodeBuild build deve eseguire i comandi buildspec durante la build del runner self-hosted di Buildkite, utilizzate la seguente sintassi:

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

Di seguito è riportato un esempio di pipeline Buildkite con un override buildspec:

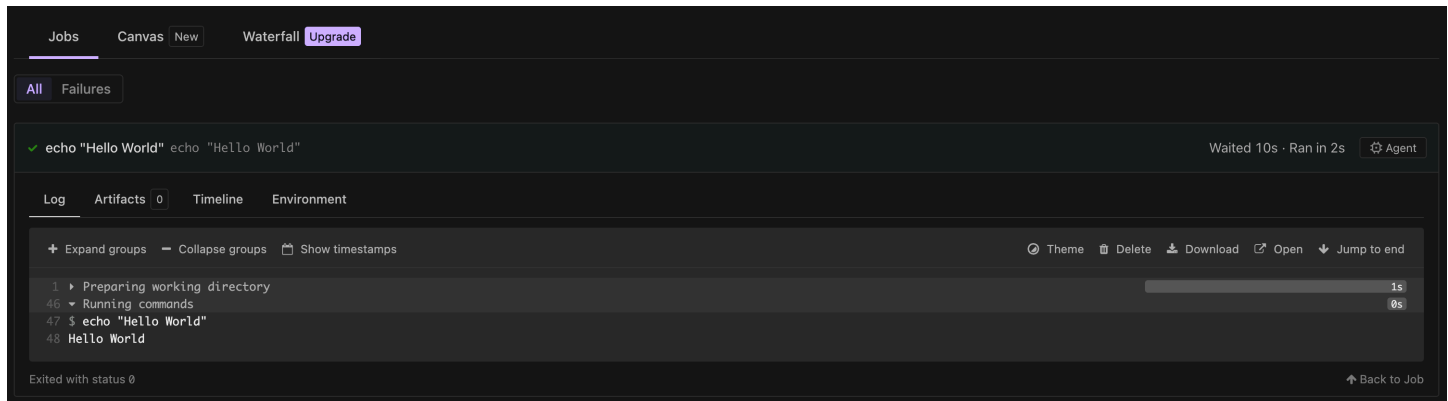
```
agents:
  project: "codebuild-myProject"
  buildspec-override: "true"
steps:
  - command: "echo \"Hello World\""
```

4. Facoltativamente, puoi fornire etichette diverse da quelle supportate. CodeBuild Queste etichette verranno ignorate allo scopo di sovrascrivere gli attributi della build, ma non falliranno la richiesta del webhook. Ad esempio, l'aggiunta myLabel: "testLabel" come etichetta non impedirà l'esecuzione della build.

Passaggio 5: rivedi i risultati

Ogni volta che viene avviato un job Buildkite nella tua pipeline, CodeBuild riceverà un evento webhook tramite il `job.scheduled` webhook Buildkite. Per ogni job della tua build di Buildkite, CodeBuild inizierà una build per far funzionare un temporaneo Buildkite runner. Il runner è responsabile dell'esecuzione di un singolo job Buildkite. Una volta completato il lavoro, il runner e il processo di compilazione associato verranno immediatamente interrotti.

Per visualizzare i registri dei lavori del flusso di lavoro, accedi alla pipeline di Buildkite e seleziona la build più recente (puoi attivare una nuova build scegliendo Nuova build). Una volta avviata la CodeBuild build associata a ciascuno dei tuoi lavori e dopo aver avviato il lavoro, dovresti vedere i log relativi al lavoro all'interno della console Buildkite



Autenticazione di Buildkite su un repository privato

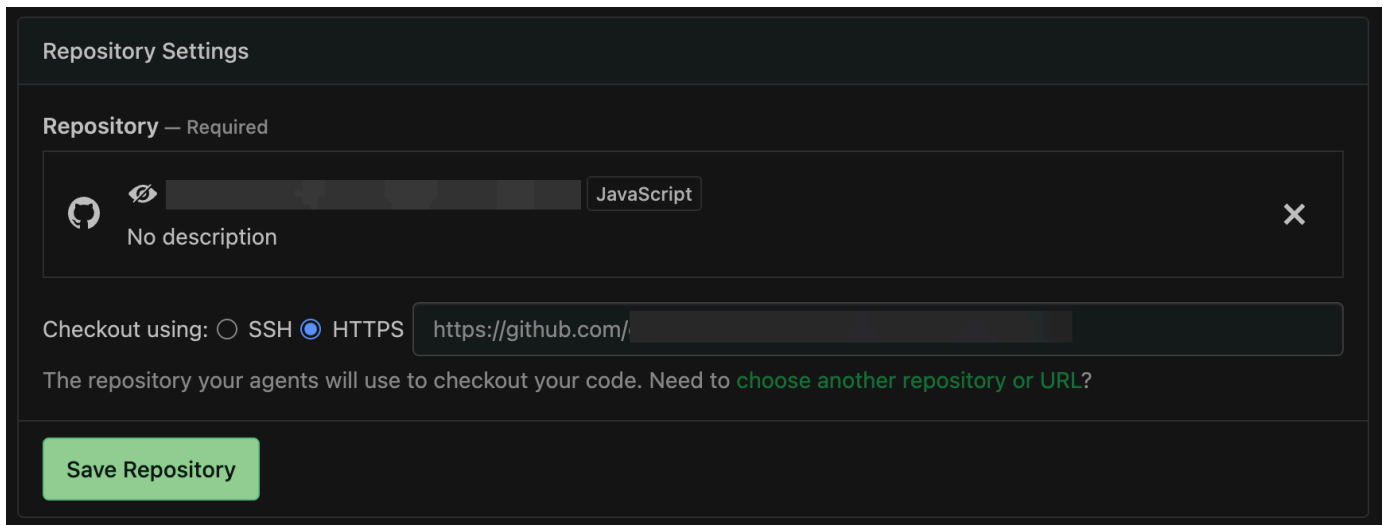
Se hai un repository privato configurato all'interno della tua pipeline di Buildkite, Buildkite richiede [autorizzazioni aggiuntive all'interno dell'ambiente di compilazione per estrarre il repository, poiché Buildkite non vende credenziali](#) a runner ospitati autonomamente perché recuperino da repository privati. Per autenticare l'agente runner self-hosted di Buildkite nel tuo repository di sorgenti private esterne, puoi utilizzare una delle seguenti opzioni.

Per autenticarsi con CodeBuild

CodeBuild offre la gestione gestita delle credenziali per i tipi di sorgenti supportati. Per utilizzare le credenziali di CodeBuild origine per estrarre l'archivio dei sorgenti del lavoro, è possibile utilizzare i seguenti passaggi:

1. Nella CodeBuild console, accedi a Modifica progetto o crea un nuovo CodeBuild progetto utilizzando la procedura descritta in [Fase 2: Creare un CodeBuild progetto con un webhook](#)
2. In Opzioni relative alle credenziali di origine di Buildkite, seleziona il provider del repository di origine del tuo lavoro.
 1. Se desideri utilizzare CodeBuild credenziali a livello di account, verifica che siano configurate correttamente. Inoltre, se il tuo progetto ha un builds spec in linea configurato, verifica che sia abilitato. [git-credential-helper](#)
 2. Se desideri utilizzare credenziali a livello di progetto, seleziona Usa CodeBuild credenziali di sostituzione solo per questo progetto e imposta le credenziali per il tuo progetto.

3. Nelle impostazioni della pipeline Buildkite, vai a Impostazioni del repository. Imposta le impostazioni di check-out del repository di origine su Checkout usando HTTPS



Per autenticarsi con i segreti di Buildkite

Buildkite mantiene un [plugin ssh-checkout](#) che può essere usato per autenticare il runner self-hosted in un repository di sorgenti esterno usando una chiave ssh. Il valore della chiave viene memorizzato come [segreto Buildkite e recuperato automaticamente dall'agente runner self-hosted di Buildkite](#) quando si tenta di aprire un repository privato. Per configurare il plugin ssh-checkout per la tua pipeline Buildkite, puoi utilizzare i seguenti passaggi:

1. Genera una chiave ssh privata e pubblica usando il tuo indirizzo email, ad es. `ssh-keygen -t rsa -b 4096 -C "myEmail@address.com"`
2. Aggiungi la chiave pubblica al tuo archivio sorgente privato. Ad esempio, puoi seguire [questa guida](#) per aggiungere una chiave a un GitHub account.
3. Aggiungi una [nuova chiave SSH segreta](#) al tuo cluster Buildkite. All'interno del tuo cluster Buildkite, seleziona Segreti → Nuovo segreto. Aggiungi un nome per il tuo segreto nel campo Chiave e aggiungi la tua chiave SSH privata nel campo Valore:

New Secret

Key — Required

SOURCE_SSH_KEY

Keys are case insensitive, can only contain alphanumeric and underscore characters, and can't start with BUILDKITE or BK

Value — Required

-----BEGIN OPENSSH PRIVATE KEY-----

4. All'interno della tua pipeline Buildkite, accedi alle impostazioni del tuo repository e imposta il checkout per utilizzare SSH.

Repository Settings

Repository — Required

  No description 

Checkout using: ☒ SSH ☐ HTTPS

The repository your agents will use to checkout your code. Need to [choose another repository or URL?](#)

[Save Repository](#)

5. Aggiorna i passaggi YAML della pipeline per utilizzare il plug-in. `git-ssh-checkout` Ad esempio, il seguente file YAML della pipeline utilizza l'azione di checkout con la precedente chiave segreta Buildkite:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "npm run build"
    plugins:
      - git-ssh-checkout#v0.4.1:
```



```
ssh-secret-key-name: 'SOURCE_SSH_KEY'
```

- Quando si esegue un job di runner self-hosted di Buildkite all'interno CodeBuild, Buildkite ora utilizzerà automaticamente il valore segreto configurato durante l'estrazione del repository privato

Opzioni di configurazione di Runner

È possibile specificare le seguenti variabili di ambiente nella configurazione del progetto per modificare la configurazione di installazione dei runner ospitati autonomamente:

- CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN:** CodeBuild recupererà il valore segreto configurato come valore di questa variabile di ambiente da per registrare il Gestione dei segreti AWS runner agent self-hosted Buildkite. Questa variabile di ambiente deve essere di tipo **SECRETS_MANAGER** e il valore deve essere il nome del segreto in Secrets Manager. Una variabile di ambiente del token Buildkite Agent è richiesta per tutti i progetti Buildkite runner.
- CODEBUILD_CONFIG_BUILDKITE_CREDENTIAL_DISABLE:** Per impostazione predefinita, CodeBuild caricherà le credenziali di origine a livello di account o progetto nell'ambiente di compilazione, poiché queste credenziali vengono utilizzate dall'agente Buildkite per estrarre l'archivio di origine del lavoro. Per disabilitare questo comportamento, puoi aggiungere questa variabile di ambiente al tuo progetto con il valore impostato su `true`, che impedirà il caricamento delle credenziali di origine nell'ambiente di compilazione.

Esegui i comandi buildspec per le fasi **INSTALL**, **PRE_BUILD** e **POST_BUILD**

Per impostazione predefinita, CodeBuild ignora qualsiasi comando buildspec quando si esegue una build Buildkite runner ospitata autonomamente. Per eseguire i comandi buildspec durante la compilazione,

```
buildspec-override: "true"
```

può essere aggiunto come suffisso all'etichetta:

```
agents:  
  project: "codebuild-<project name>"  
  buildspec-override: "true"
```

Utilizzando questo comando, CodeBuild creerà una cartella chiamata `buildkite-runner` nella cartella di origine principale del contenitore. Quando il runner Buildkite si avvia durante la BUILD fase, il runner verrà eseguito nella directory. `buildkite-runner`

Esistono diverse limitazioni quando si utilizza un override di `buildspec` in una build Buildkite ospitata autonomamente:

- L'agente Buildkite richiede che le credenziali di origine esistano all'interno dell'ambiente di compilazione per recuperare l'archivio dei sorgenti del lavoro. Se utilizzi le credenziali CodeBuild di origine per l'autenticazione, dovrai abilitarle nel tuo `buildspec`. `git-credential-helper`
Ad esempio, puoi usare il seguente `buildspec` da abilitare per le tue build Buildkite: `git-credential-helper`

```
version: 0.2
env:
  git-credential-helper: yes
phases:
  pre_build:
    commands:
      - echo "Hello World"
```

- CodeBuild non eseguirà i comandi `buildspec` durante la fase, poiché il runner self-hosted viene eseguito durante la BUILD fase. BUILD
- CodeBuild non supporta i file `buildspec` per le build di Buildkite runner. Per i runner self-hosted di Buildkite sono supportate solo le specifiche di build in linea
- Se un comando build fallisce nella INSTALL fase PRE_BUILD or, non CodeBuild avvierà il runner self-hosted e il job Buildkite dovrà essere annullato manualmente.

Configurazione di un Buildkite runner a livello di codice

Per configurare un progetto Buildkite runner in modo programmatico, dovrai configurare le seguenti risorse:

Per creare un Buildkite runner a livello di codice

1. Crea un token dell'agente Buildkite e salva il token in testo semplice all'interno. Gestione dei segreti AWS
2. Configura un CodeBuild progetto con la tua configurazione preferita. Dovrai configurare i seguenti attributi aggiuntivi:

1. Un valore di ambiente con nome `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKENSECRETS_MANAGER`, tipo e un valore uguale al token dell'agente Buildkite associato al cluster Buildkite.
2. Tipo di sorgente uguale a `NO_SOURCE`
3. Autorizzazioni per accedere al segreto creato nella fase 1 del ruolo di servizio del progetto

Ad esempio, puoi utilizzare il seguente comando per creare un progetto Buildkite runner valido tramite la CLI:

```
aws codebuild create-project \
--name buildkite-runner-project \
--source "{\"type\": \"NO_SOURCE\", \"buildspec\": \"\"}" \
--environment "{\"image\": \"aws/codebuild/amazonlinux-x86_64-standard:5.0\", \"type\": \"LINUX_CONTAINER\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\", \"environmentVariables\": [{\"name\": \"CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN\", \"type\": \"SECRETS_MANAGER\", \"value\": \"<buildkite-secret-name>\"}]}" \
--artifacts "{\"type\": \"NO_ARTIFACTS\"}" \
--service-role <service-role>
```

3. Crea un webhook Buildkite runner sul progetto creato nel passaggio 2. Dovrai utilizzare le seguenti opzioni di configurazione durante la creazione del webhook:
 1. build-type deve essere uguale a `RUNNER_BUILDKITE_BUILD`
 2. Un filtro con tipo `EVENT` e pattern uguali a `WORKFLOW_JOB_QUEUED`

Ad esempio, puoi utilizzare il seguente comando per creare un webhook Buildkite runner valido tramite la CLI:

```
aws codebuild create-webhook \
--project-name buildkite-runner-project \
--filter-groups "[[{\"type\": \"EVENT\", \"pattern\": \"WORKFLOW_JOB_QUEUED\"}]]" \
--build-type RUNNER_BUILDKITE_BUILD
```

4. Salva l'URL di payload e i valori Secret restituiti dalla `create-webhook` chiamata e usa le credenziali per creare un webhook all'interno della console Buildkite. Puoi fare riferimento al [Passaggio 3: Crea un CodeBuild webhook all'interno di Buildkite](#) [Tutorial: configura un CodeBuild Buildkite runner ospitato](#) per una guida su come configurare questa risorsa.

Risolvi i problemi relativi al webhook per le build fallite o per un lavoro sospeso

Problema:

Il webhook che hai configurato [Tutorial: configura un CodeBuild Buildkite runner ospitato](#) non funziona o il tuo processo di workflow è bloccato in Buildkite.

Possibili cause:

- Il tuo evento webhook `job.scheduled` potrebbe non riuscire ad attivare una build. Esamina i registri delle risposte per visualizzare la risposta o il messaggio di errore.
- La tua CodeBuild build fallisce prima di avviare l'agente runner self-hosted di Buildkite per gestire il tuo lavoro.

Soluzioni consigliate:

Per eseguire il debug degli eventi webhook Buildkite non riusciti:

1. Nelle impostazioni della tua organizzazione Buildkite, vai a Notification Services, seleziona il tuo CodeBuild webhook e poi trova il registro delle richieste.
2. Trova l'evento `job.scheduled` webhook associato al tuo lavoro Buildkite bloccato. Puoi utilizzare il campo ID del lavoro all'interno del payload del webhook per correlare l'evento webhook al tuo lavoro Buildkite.
3. Seleziona la scheda Risposta e controlla il corpo della risposta. Verifica che il codice di stato della risposta sia 200 impostato e che il corpo della risposta non contenga messaggi imprevisti.

The screenshot displays the 'Request Log' interface in AWS CodeBuild. At the top, a red status box shows '400'. To its right, a dropdown menu is set to 'job.scheduled', followed by a duration of '0.29s' and a timestamp '2024-12-11 23:03:33 UTC' with a downward arrow. Below this, there are two tabs: 'Request' and 'Response', with 'Response' being the active tab. Under the 'Response' tab, there are two sections: 'Headers' and 'Body'. The 'Headers' section contains the following text: 'Date: Wed, 11 Dec 2024 23:03:33 GMT', 'Connection: close', 'Content-Type: application/json', 'Content-Length: 92', 'X-Amzn-Errortype: InvalidInputException:http://', and 'X-Amzn-Requestid: 7a931bed-0bae-4c1a-9f5b-2178900cd180'. The 'Body' section contains a JSON object: '{ "message": "Project name in label nonMatchingProjectName did not match actual project name" }'.

Risolvi i problemi di autorizzazione del webhook

Problema:

Il job Buildkite non riesce a eseguire il checkout del repository dei sorgenti del lavoro a causa di problemi di autorizzazione.

Possibili cause:

- CodeBuild non dispone di autorizzazioni sufficienti per il check-out dell'archivio dei sorgenti del lavoro.
- Le impostazioni del repository della pipeline sono impostate per il check-out utilizzando SSH per le credenziali gestite. CodeBuild

Soluzioni consigliate:

- Verifica CodeBuild che siano configurate autorizzazioni sufficienti per accedere all'archivio dei sorgenti del lavoro. Inoltre, verifica che il ruolo di servizio del CodeBuild progetto disponga di autorizzazioni sufficienti per accedere all'opzione di autorizzazione di origine configurata.
- Verifica che la tua pipeline Buildkite sia configurata per utilizzare il checkout tramite HTTPS se utilizzi credenziali del repository di origine gestito. CodeBuild

Le sostituzioni delle etichette sono supportate con il runner Buildkite ospitato CodeBuild

Nella tua pipeline Buildkite Steps Agent Tag Labels, puoi fornire una serie di sostituzioni di etichette che modificano la build del runner ospitato autonomamente. Tutte le build non riconosciute da CodeBuild verranno ignorate ma non falliranno la richiesta del webhook. Ad esempio, il seguente flusso di lavoro YAML include sostituzioni per immagine, dimensione dell'istanza, flotta e buildspec:

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
      buildspec-override: "true"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "al2-5.0"
        size:
          - "small"
          - "large"
```

project:codebuild-*<project-name>*(richiesto)

- Ad esempio: project: "codebuild-myProject"
- Obbligatorio per tutte le configurazioni delle fasi della pipeline di Buildkite. *<project name>* deve essere uguale al nome del progetto per il quale è configurato il webhook runner self-hosted.

queue: "*<queue-name>*"

- Ad esempio: queue: "*<queue-name>*"
- Utilizzato per indirizzare i lavori Buildkite a una coda specifica. Vedi il [Buildkite Agent Queue](#) Tag per ulteriori informazioni.

image: "*<environment-type>-<image-identifier>*"

- Ad esempio: `image: "arm-3.0"`
 - Sostituisce l'immagine e il tipo di ambiente utilizzati quando si avvia la build self-hosted runner con un'immagine curata. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner Buildkite ospitato CodeBuild](#)
1. Per sovrascrivere l'immagine e il tipo di ambiente utilizzati con un'immagine personalizzata, usa `image: "custom-<environment-type>-<custom-image-identifier>"`
 2. Esempio:

```
image:
  "custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0"
```

Note

Se l'immagine personalizzata si trova in un registro privato, è necessario configurare le credenziali di registro appropriate nel progetto. CodeBuild

`instance-size: "<instance-size>"`

- Ad esempio: `instance-size: "medium"`
- Sostituisce il tipo di istanza utilizzato all'avvio della build self-hosted runner. Per maggiori informazioni sui valori supportati, consulta. [Immagini di calcolo supportate con il runner Buildkite ospitato CodeBuild](#)

`fleet: "<fleet-name>"`

- Ad esempio: `fleet: "myFleet"`
- Sostituisce le impostazioni del parco veicoli configurate nel progetto per utilizzare il parco veicoli specificato. Per ulteriori informazioni, consulta [Run si basa su flotte a capacità riservata](#).

`buildspec-override: "<boolean>"`

- Ad esempio: `buildspec-override: "true"`
- Consente alla build di eseguire i comandi buildspec nelle fasi, e INSTALL PRE_BUILDPOST_BUILD, se impostata su. true

Immagini di calcolo supportate con il runner Buildkite ospitato CodeBuild

Nell'etichetta in cui hai configurato [Buildkite runner autogestito in AWS CodeBuild](#), puoi sovrascrivere le impostazioni dell'ambiente Amazon EC2 utilizzando i valori nelle prime tre colonne. CodeBuild fornisce le seguenti immagini di calcolo di Amazon EC2. Per ulteriori informazioni sull'

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
linux	4.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-standard:4.0	al/standard/4.0
		medium			
		large			
		xlarge			
linux	5.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-standard:5.0	al/standard/5.0
		gpu_small			
		gpu_large			
linux-ec2	latest	small	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-x86_64-base:latest	Nessuno
		medium			
		large			
arm	2.0	small	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
		medium			
		large			
		xlarge			
		2xlarge			

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
arm	3.0		Amazon Linux 2023	aws/codebuild/amazonlinux-arm64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Nessuno
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge 2xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/D

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza	Platform (Piattaforma)	Immagine risolta	Definizione
			Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/D
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/D
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/D
windows-ec2	2022	medium large	Windows Server Core 2022	aws/codebuild/ami/windows-base:2022	Nessuno

Inoltre, puoi sovrascrivere le impostazioni dell'ambiente Lambda utilizzando i seguenti valori. Per ulteriori informazioni sul calcolo CodeBuild Lambda, vedere. [Esegui build su calcolo AWS Lambda](#)
CodeBuild supporta le seguenti immagini di calcolo Lambda:

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto1	4GB			
	1	8GB			

Tipo di ambiente	Identificatore dell'immagine	Dimensioni istanza			
	corretto17	10GB			
	corretto21				
	nodejs18				
	nodejs20				
	python3.11				
	python3.12				
	ruby3.2				

Per ulteriori informazioni, consultare [Modi e tipi di calcolo dell'ambiente di creazione](#) e [Immagini Docker fornite da CodeBuild](#).

Usa i webhook con AWS CodeBuild

AWS CodeBuild supporta l'integrazione dei webhook con GitHub GitHub Enterprise Server GitLab, GitLab Self Managed e Bitbucket.

Argomenti

- [Le migliori pratiche per l'utilizzo dei webhook con AWS CodeBuild](#)
- [Eventi webhook Bitbucket](#)
- [GitHub webhook globali e organizzativi](#)
- [GitHub webhook manuali](#)
- [GitHub eventi webhook](#)
- [GitLab webhook di gruppo](#)
- [GitLab webhook manuali](#)

- [GitLab eventi webhook](#)
- [Webhook manuali Buildkite](#)
- [Pull richiedi l'approvazione dei commenti](#)

Le migliori pratiche per l'utilizzo dei webhook con AWS CodeBuild

Per i progetti che utilizzano archivi pubblici per configurare i webhook, consigliamo le seguenti opzioni:

Filtri di configurazione ACTOR_ACCOUNT_ID

Aggiungi ACTOR_ACCOUNT_ID filtri ai gruppi di filtri webhook del tuo progetto per specificare quali utenti possono attivare una build. Ogni evento webhook inviato a CodeBuild viene fornito con informazioni sul mittente che specificano l'identificatore dell'attore. CodeBuild filtrerà i webhook in base al modello di espressione regolare fornito nei filtri. Puoi specificare gli utenti specifici autorizzati ad attivare le build con questo filtro. Per ulteriori informazioni, consultare [GitHub eventi webhook](#) e [Eventi webhook Bitbucket](#).

Imposta FILE_PATH i filtri

Aggiungi FILE_PATH filtri ai gruppi di filtri webhook del tuo progetto per includere o escludere i file che possono attivare una build quando vengono modificati. Ad esempio, puoi negare le richieste di compilazione di modifiche al `buildspec.yml` file utilizzando un modello di espressione regolare come `^buildspec.yml$`, insieme alla `excludeMatchedPattern` proprietà. Per ulteriori informazioni, consultare [GitHub eventi webhook](#) e [Eventi webhook Bitbucket](#).

Definisci le autorizzazioni per il tuo ruolo di build IAM

Le build attivate da un webhook utilizzano il ruolo del servizio IAM specificato nel progetto. Ti consigliamo di impostare le autorizzazioni nel ruolo di servizio sul set minimo di autorizzazioni richieste per eseguire la build. Ad esempio, in uno scenario di test e distribuzione, crea un progetto per il test e un altro progetto per la distribuzione. Il progetto di test accetta build di webhook dal repository, ma non fornisce autorizzazioni di scrittura per le tue risorse. Il progetto di distribuzione fornisce autorizzazioni di scrittura per le risorse e il filtro webhook è configurato per consentire solo agli utenti attendibili di attivare le build.

Usa una specifica di build archiviata in linea o Amazon S3

Se definisci il tuo `buildspec` in linea all'interno del progetto stesso o memorizzi il file `buildspec` in un bucket Amazon S3, il file `buildspec` è visibile solo al proprietario del progetto. Ciò impedisce

alle pull request di apportare modifiche al codice del file `buildspec` e di attivare build indesiderate. Per ulteriori informazioni, consulta [ProjectSource.buildspec](#) nell'API Reference. CodeBuild

Eventi webhook Bitbucket

È possibile utilizzare gruppi di filtri di webhook per specificare quali eventi webhook Bitbucket attivano una compilazione. Ad esempio, puoi specificare che una build venga attivata solo per le modifiche a rami specifici.

È possibile creare uno o più gruppi di filtri webhook per specificare quali eventi webhook attivano una build. Una build viene attivata se un gruppo di filtri restituisce `true`, il che si verifica quando tutti i filtri del gruppo restituiscono `true`. Quando si crea un gruppo di filtri, si specifica:

Un evento

Per Bitbucket, puoi scegliere uno o più dei seguenti eventi:

- `PUSH`
- `PULL_REQUEST_CREATED`
- `PULL_REQUEST_UPDATED`
- `PULL_REQUEST_MERGED`
- `PULL_REQUEST_CLOSED`

Il tipo di evento del webhook è nell'intestazione del campo `X-Event-Key`. La tabella seguente mostra il modo in cui i valori dell'intestazione `X-Event-Key` sono mappati ai tipi di eventi.

Note

È necessario abilitare l'evento `merged` evento nell'impostazione del webhook Bitbucket se viene creato un gruppo di filtri di webhook che utilizza il tipo di evento `PULL_REQUEST_MERGED`. È inoltre necessario abilitare l'evento `declined` nelle impostazioni del webhook di Bitbucket se si crea un gruppo di filtri webhook che utilizza il tipo di evento. `PULL_REQUEST_CLOSED`

Valore intestazione X-Event-Key	Tipo di evento
<code>repo:push</code>	<code>PUSH</code>

Valore intestazione X-Event-Key	Tipo di evento
<code>pullrequest:created</code>	<code>PULL_REQUEST_CREATED</code>
<code>pullrequest:updated</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pullrequest:fulfilled</code>	<code>PULL_REQUEST_MERGED</code>
<code>pullrequest:rejected</code>	<code>PULL_REQUEST_CLOSED</code>

Infatti `PULL_REQUEST_MERGED`, se una pull request viene unita alla strategia squash e il ramo pull request viene chiuso, il commit originale della pull request non esiste più. In questo caso, la variabile di `CODEBUILD_WEBHOOK_MERGE_COMMIT` ambiente contiene l'identificatore dello squashed merge commit.

Uno o più filtri opzionali

Usa un'espressione regolare per specificare un filtro. Affinché un evento attivi una build, ogni filtro all'interno del gruppo ad esso associato deve restituire `true`.

`ACTOR_ACCOUNT_ID(ACTOR_ID`nella console)

Un evento webhook attiva una build quando l'ID di un account Bitbucket corrisponde al modello di espressione regolare. Questo valore è presente nella proprietà `account_id` dell'oggetto `actor` nel payload del filtro webhook.

`HEAD_REF`

Un evento webhook attiva una build quando il riferimento alla testina corrisponde al modello di espressione regolare (ad esempio, `and`). `refs/heads/branch-name` `refs/tags/tag-name` Un filtro `HEAD_REF` valuta il nome di riferimento Git per il ramo o il tag. Il nome del ramo o del tag è presente nel campo `name` dell'oggetto `new` nell'oggetto `push` del payload del webhook. Per gli eventi di richiesta pull, il nome del ramo è presente nel campo `name` dell'oggetto `branch` dell'oggetto `source` nel payload del webhook.

`BASE_REF`

Un evento webhook attiva una build quando il riferimento di base corrisponde al modello di espressione regolare. Un filtro `BASE_REF` funziona solo con gli eventi di richiesta pull (ad esempio, `refs/heads/branch-name`). Un filtro `BASE_REF` valuta il nome di riferimento Git per il ramo. Il nome del ramo è presente nel campo `name` dell'oggetto `branch` nell'oggetto `destination` nel payload del webhook.

FILE_PATH

Un webhook attiva una build quando il percorso di un file modificato corrisponde al modello di espressione regolare.

COMMIT_MESSAGE

Un webhook attiva una build quando il messaggio di head commit corrisponde al modello di espressione regolare.

WORKFLOW_NAME

Un webhook attiva una build quando il nome del flusso di lavoro corrisponde al modello di espressione regolare.

Note

Puoi trovare il payload del webhook nelle impostazioni del webhook del tuo repository Bitbucket.

Argomenti

- [Filtro di eventi webhook Bitbucket \(console\)](#)
- [Filtro di eventi webhook Bitbucket \(SDK\)](#)
- [Filtro di eventi webhook Bitbucket \(CloudFormation\)](#)

Filtro di eventi webhook Bitbucket (console)

Per utilizzare il per Console di gestione AWS filtrare gli eventi del webhook:

1. Al momento della creazione di un progetto, selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository).
2. Da Event type (Tipo di evento), selezionare uno o più eventi.
3. Per applicare un filtro che stabilisce quando un evento avvia una compilazione, in Start a build under these conditions (Avvia una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.

4. Per applicare un filtro che stabilisce quando un evento non avvia una compilazione, in **Don't start a build under these conditions** (Non avviare una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
5. Per aggiungere un altro gruppo di filtri, scegliere **Add filter group** (Aggiungi gruppo di filtri).

Per ulteriori informazioni, consulta le pagine [Creare un progetto di compilazione \(console\)](#) e [WebhookFilter](#) nella Documentazione di riferimento dell'API AWS CodeBuild .

In questo esempio, un gruppo di filtri webhook avvia una compilazione solo per le richieste pull:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕**PULL_REQUEST_UPDATED** ✕**PULL_REQUEST_MERGED** ✕**PULL_REQUEST_CLOSED** ✕

► **Start a build under these conditions - optional**

► **Don't start a build under these conditions - optional**

In un esempio con due gruppi di filtri, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create o aggiornate su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/branch1!`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook avvia una compilazione per tutte le richieste pull, tranne gli eventi tag:

Filter group 1

Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

► Start a build under these conditions - optional

▼ Don't start a build under these conditions - optional

Add filter

Filter 1

Type

HEAD_REF ▼

Pattern

^refs/tags/.*

In questo esempio, un gruppo di filtri di webhook attiva una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook attiva una build solo quando i file vengono modificati nelle cartelle o nelle cartelle. `src test`

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook avvia una compilazione solo quando un utente Bitbucket che non dispone di un ID account corrispondente all'espressione regolare `actor-account-id` apporta una modifica.

 Note

Per informazioni su come trovare l'ID del tuo account Bitbucket, consulta <https://api.bitbucket.org/2.0/users/user-name>, `user-name` dov'è il tuo nome utente Bitbucket.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

In questo esempio, un gruppo di filtri di webhook attiva una compilazione per un evento push quando il messaggio di commit HEAD corrisponde all'espressione regolare `\[CodeBuild\]`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Filtro di eventi webhook Bitbucket (SDK)

Per utilizzare l' AWS CodeBuild SDK per filtrare gli eventi webhook, utilizza il `filterGroups` campo nella sintassi di richiesta dei metodi o API. `CreateWebhook` `UpdateWebhook` Per ulteriori informazioni, consulta [WebhookFilter](#) nella documentazione di riferimento dell'API CodeBuild .

Per creare un filtro di webhook che attivi una compilazione solo per le richieste pull, inserire nella sintassi di richiesta quanto segue:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    }
  ]
]
```

Per creare un filtro di webhook che attivi una compilazione solo per determinati rami, utilizzare il parametro `pattern` per specificare un'espressione regolare che filtri i nomi dei rami. In un esempio con due gruppi di filtri, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create o aggiornate su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/myBranch$`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Il parametro `excludeMatchedPattern` consente di indicare quali eventi non attivano una compilazione. In questo esempio, viene attivata una compilazione per tutte le richieste, tranne gli eventi tag.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```

```
    "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
  },
  {
    "type": "HEAD_REF",
    "pattern": "^refs/tags/.*",
    "excludeMatchedPattern": true
  }
]
]
```

È possibile creare un filtro che attivi una compilazione solo quando un utente Bitbucket che dispone di un ID account `actor-account-id` apporta una modifica.

Note

Per informazioni su come trovare l'ID del tuo account Bitbucket, consulta <https://api.bitbucket.org/2.0/users/user-name>, *user-name* dov'è il tuo nome utente Bitbucket.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

È possibile creare un filtro che attivi una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare nell'argomento `pattern`. In questo esempio, il gruppo di filtri indica che una compilazione si attiva solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

```
"filterGroups": [
  [
```

```
{
  "type": "EVENT",
  "pattern": "PUSH"
},
{
  "type": "FILE_PATH",
  "pattern": "^buildspec.*"
}
]
```

In questo esempio, il gruppo di filtri specifica che una build viene attivata solo quando i file vengono modificati nelle cartelle o nelle cartelle. `src test`

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

È possibile creare un filtro che attiva una compilazione solo quando il messaggio di commit HEAD corrisponde all'espressione regolare nell'argomento del modello. In questo esempio, il gruppo di filtri specifica che viene attivata una compilazione solo quando il messaggio di commit HEAD dell'evento push corrisponde all'espressione regolare `\[CodeBuild\]`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "COMMIT_MESSAGE",
      "pattern": "\[CodeBuild\]"
    }
  ]
]
```



```
]
]
```

Filtro di eventi webhook Bitbucket (CloudFormation)

Per utilizzare un CloudFormation modello per filtrare gli eventi webhook, utilizzate la proprietà del AWS CodeBuild progetto. `FilterGroups` La parte di un modello CloudFormation in formato YAML riportata di seguito crea due gruppi di filtri. Insieme, questi attivano una compilazione quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create o aggiornate su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` da un utente Bitbucket che non dispone di un ID account 12345.
- Il secondo gruppo di filtri specifica le richieste push create su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/.*`.
- Il terzo gruppo di filtri specifica una richiesta push con un messaggio di commit HEAD corrispondente all'espressione regolare `\[CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
```

- Type: ACTOR_ACCOUNT_ID
Pattern: 12345
ExcludeMatchedPattern: true
- - Type: EVENT
Pattern: PUSH
- Type: HEAD_REF
Pattern: ^refs/heads/.*
- Type: FILE_PATH
Pattern: READ_ME
ExcludeMatchedPattern: true
- - Type: EVENT
Pattern: PUSH
- Type: COMMIT_MESSAGE
Pattern: \[CodeBuild\]
- Type: FILE_PATH
Pattern: ^src/.+|^test/.+

GitHub webhook globali e organizzativi

È possibile utilizzare i webhook CodeBuild GitHub globali o organizzativi per avviare build su eventi webhook da qualsiasi repository all'interno di un'organizzazione o di un'azienda. GitHub I webhook globali e organizzativi funzionano con tutti i tipi di eventi GitHub webhook esistenti e possono essere configurati aggiungendo una configurazione di ambito durante la creazione di un webhook. CodeBuild Puoi anche utilizzare webhook globali e organizzativi per [configurare GitHub Action runner ospitati autonomamente al fine di ricevere WORKFLOW_JOB_QUEUED eventi da più repository CodeBuild all'interno](#) di un singolo progetto.

Argomenti

- [Configura un webhook globale o organizzativo GitHub](#)
- [Filtra gli eventi webhook GitHub globali o organizzativi \(console\)](#)
- [Filtra gli eventi webhook GitHub dell'organizzazione \(\)CloudFormation](#)

Configura un webhook globale o organizzativo GitHub

I passaggi principali per configurare un GitHub webhook globale o organizzativo sono i seguenti. Per ulteriori informazioni sui GitHub webhook globali e organizzativi, vedere. [GitHub webhook globali e organizzativi](#)

1. Imposta la posizione di origine del progetto su.
`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`
2. Nella configurazione dell'ambito del webhook, imposta l'ambito su uno `GITHUB_ORGANIZATION` o in `GITHUB_GLOBAL` base al fatto che si tratti di un'organizzazione o di un [webhook globale](#). Per ulteriori informazioni, consulta [Tipi di webhook](#).
3. Specificate un nome come parte della configurazione dell'ambito del webhook. Per i webhook dell'organizzazione, questo è il nome dell'organizzazione, mentre per i webhook globali è il nome dell'azienda.

 Note

Se il tipo di sorgente del progetto è `GITHUB_ENTERPRISE`, sarà inoltre necessario specificare un dominio come parte della configurazione dell'ambito del webhook.

4. (Facoltativo) Se desideri ricevere solo eventi webhook per repository specifici all'interno della tua organizzazione o azienda, puoi specificarli `REPOSITORY_NAME` come filtro durante la creazione del webhook.
5. Se state creando un webhook aziendale, assicuratevi di disporre delle autorizzazioni necessarie per creare webhook a livello di organizzazione all'interno. CodeBuild GitHub Puoi creare un token di accesso GitHub personale con le autorizzazioni dei webhook dell'organizzazione o utilizzarlo. CodeBuild OAuth Per ulteriori informazioni, consulta [GitHub e token di accesso a GitHub Enterprise Server](#).

Tieni presente che i webhook dell'organizzazione funzionano con tutti i tipi di eventi webhook esistenti GitHub .

6. Se stai creando un webhook globale, il webhook dovrà essere creato manualmente. Per ulteriori informazioni su come creare manualmente un webhook all'interno GitHub, consulta. [GitHub webhook manuali](#)

Tieni presente che i webhook globali supportano solo il `WORKFLOW_JOB_QUEUED` tipo di evento. Per ulteriori informazioni, consulta [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#).

Filtra gli eventi webhook GitHub globali o organizzativi (console)

Quando crei un GitHub progetto tramite la console, seleziona le seguenti opzioni per creare un webhook GitHub globale o organizzativo all'interno del progetto. Per ulteriori informazioni sui GitHub webhook globali e organizzativi, consulta. [GitHub webhook globali e organizzativi](#)

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
 - In Source (Origine):
 - Come fornitore di origine, scegli GitHubEnterprise. GitHub
 - Per Repository, scegli webhook GitHubcon ambito.

Il GitHub repository verrà impostato automaticamente su `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, che è la posizione di origine richiesta per i webhook globali e organizzativi.

Note

Se utilizzi webhook a livello di organizzazione, assicurati che disponga delle autorizzazioni necessarie per creare webhook a CodeBuild livello di organizzazione all'interno. GitHub Se utilizzi una [OAuthconnessione esistente](#), potrebbe essere necessario rigenerare la connessione per concedere questa autorizzazione. CodeBuild In alternativa, puoi creare il webhook manualmente utilizzando la funzione [webhook CodeBuild manuale](#). Tieni presente che se disponi di un GitHub OAuth token esistente e desideri aggiungere ulteriori autorizzazioni all'organizzazione, puoi [revocare l'autorizzazione del OAuth token e ricollegare il token](#) tramite la console. CodeBuild

Source**Add source****Source 1 - Primary**

Source provider

GitHub ▼

Repository

☐ Repository in my GitHub account☐ Public repository☒ GitHub scoped webhook

GitHub repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

Connection status

You are connected to GitHub using a personal access token.

Disconnect from GitHub

- Negli eventi del webhook della fonte primaria:
 - Per il tipo di ambito, scegli il livello di organizzazione se stai creando un webhook aziendale o il livello Enterprise se stai creando un webhook globale.
 - Per Nome, inserisci il nome dell'azienda o dell'organizzazione, a seconda che il webhook sia un webhook globale o dell'organizzazione.

Se il tipo di origine del progetto è `GITHUB_ENTERPRISE`, è inoltre necessario specificare un dominio come parte della configurazione dell'organizzazione del webhook. Ad esempio, se l'URL della tua organizzazione è `https://domain.com/orgs/org-name`, allora il dominio è `https://domain.com`.

Note

Questo nome non può essere modificato dopo la creazione del webhook. Per modificare il nome, è possibile eliminare e ricreare il webhook. Se desideri rimuovere completamente il webhook, puoi anche aggiornare la posizione di origine del progetto in un repository. GitHub

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#) ☒ Rebuild every time a code change is pushed to this repository

Scope type

☒ Organization level☐ Enterprise level

Organization name

Your GitHub organization name.

Build type

☒ Single build
Triggers single build☐ Batch build
Triggers multiple builds as single execution► **Additional configuration**

- (Facoltativo) Nei gruppi di filtri degli eventi Webhook, puoi specificare quali [eventi desideri attivare una](#) nuova build. È inoltre possibile specificare REPOSITORY_NAME come filtro di attivare solo le build sugli eventi webhook provenienti da repository specifici.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

REPOSITORY_NAME ▼

Pattern

repository-name

[Remove](#)

Puoi anche impostare il tipo di evento su per configurare gli Actions runner WORKFLOW_JOB_QUEUED ospitati autonomamente. GitHub Per ulteriori informazioni, consulta [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#).

3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.

Filtra gli eventi webhook GitHub dell'organizzazione ()CloudFormation

Per utilizzare un CloudFormation modello per filtrare gli eventi del webhook dell'organizzazione, utilizzate la proprietà del AWS CodeBuild ScopeConfiguration progetto. Per ulteriori informazioni sui GitHub webhook globali e organizzativi, consulta. [GitHub webhook globali e organizzativi](#)

Note

I webhook globali e i webhook GitHub Enterprise non sono supportati da. CloudFormation

La seguente parte di un modello in formato YAML crea quattro gruppi di filtri. CloudFormation Insieme, attivano una build quando uno o tutti restituiscono true:

- Il primo gruppo di filtri specifica che le richieste pull vengono create o aggiornate sui rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` da un GitHub utente che non dispone di un ID 12345 account.
- Il secondo gruppo di filtri specifica le richieste push create su file con nomi che corrispondono all'espressione regolare `README` in rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/.*`.
- Il terzo gruppo di filtri specifica una richiesta push con un messaggio di commit HEAD corrispondente all'espressione regolare `\[CodeBuild\]`.
- Il quarto gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions con un nome di flusso di lavoro corrispondente all'espressione regolare `\[CI-CodeBuild\]`

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
      - Type: ACTOR_ACCOUNT_ID
        Pattern: 12345
        ExcludeMatchedPattern: true
      - - Type: EVENT
```



```

    Pattern: PUSH
  - Type: HEAD_REF
    Pattern: ^refs/heads/.+
  - Type: FILE_PATH
    Pattern: README
    ExcludeMatchedPattern: true
  - - Type: EVENT
    Pattern: PUSH
  - Type: COMMIT_MESSAGE
    Pattern: \[CodeBuild\]
  - Type: FILE_PATH
    Pattern: ^src/.+|^test/.+
  - - Type: EVENT
    Pattern: WORKFLOW_JOB_QUEUED
  - Type: WORKFLOW_NAME
    Pattern: \[CI-CodeBuild\]

```

GitHub webhook manuali

È possibile configurare i GitHub webhook manuali per CodeBuild impedire il tentativo automatico di creare un webhook al loro interno. GitHub CodeBuild restituisce un URL di payload come parte della chiamata per creare il webhook e può essere utilizzato per creare manualmente il webhook al suo interno. GitHub Anche se la creazione di un webhook nel tuo GitHub account non CodeBuild è consentita, puoi comunque creare manualmente un webhook per il tuo progetto di build.

Usa la seguente procedura per creare un GitHub webhook manuale.

Per creare un webhook GitHub manuale

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
 - In Source (Origine):
 - Per Source provider, scegli. GitHub
 - Per Repository, scegli Repository nel mio GitHub account.
 - Nel campo Repository URL (URL repository), inserire **https://github.com/*user-name/repository-name***
 - Negli eventi webhook di Primary Source:

- Per Webhook: facoltativo, scegli Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository.
 - Scegli Configurazione aggiuntiva e per Creazione manuale: facoltativo, scegli Crea manualmente un webhook per questo repository nella console. GitHub .
3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione. Prendi nota dei valori Payload URL e Secret poiché li utilizzerai in seguito.

Create webhook

×

You must create a webhook for your GitHub repository.

Payload URL

Copy payload URL

Secret

Copy secret

Close

4. Apri la GitHub console all'indirizzo **<https://github.com/user-name/repository-name/settings/hooks>** e scegli Aggiungi webhook.
- Per Payload URL, inserisci il valore dell'URL di payload di cui hai preso nota in precedenza.
 - Per Tipo di contenuto, scegli application/json.
 - Per Segreto, inserisci il valore Segreto di cui hai preso nota in precedenza.
 - Configura i singoli eventi a cui verrà inviato un payload del webhook. CodeBuild Per quali eventi desideri attivare questo webhook? , scegli Consentimi di selezionare singoli eventi, quindi scegli tra i seguenti eventi: Push, Pull request e Releases. Se desideri avviare le build per WORKFLOW_JOB_QUEUED eventi, scegli Workflow jobs. Per ulteriori informazioni sui runner GitHub Actions, consulta [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#) Per ulteriori informazioni sui tipi di eventi supportati da CodeBuild, consulta [GitHub eventi webhook](#).
5. Scegli Aggiungi webhook.

GitHub eventi webhook

È possibile utilizzare i gruppi di filtri webhook per specificare quali eventi GitHub webhook attivano una build. Ad esempio, è possibile specificare che una build venga attivata solo per le modifiche a rami specifici.

È possibile creare uno o più gruppi di filtri webhook per specificare quali eventi webhook attivano una build. Una build viene attivata se un gruppo di filtri restituisce true, il che si verifica quando tutti i filtri del gruppo restituiscono true. Quando si crea un gruppo di filtri, si specifica:

Un evento

Infatti GitHub, puoi scegliere uno o più dei seguenti

eventi: PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED, PULL_REQUEST_CLOSEDRELEASED, PRERELEASED, e WORKFLOW_JOB_QUEUED. Il tipo di evento webhook si trova nell'intestazione X-GitHub-Event nel payload del webhook. Nell'intestazione X-GitHub-Event puoi vedere pull_request o push. Per un evento di richiesta pull, il tipo si trova nel campo action del payload del webhook. La tabella seguente mostra il modo in cui i valori dell'intestazione X-GitHub-Event e i valori del campo action del payload della richiesta pull del webhook sono mappati ai tipi di eventi disponibili.

Valore intestazione X-GitHub-Event	Valore action del payload di eventi webhook	Tipo di evento
pull_request	opened	PULL_REQUEST_CREATED
pull_request	reopened	PULL_REQUEST_REOPENED
pull_request	synchronize	PULL_REQUEST_UPDATED
pull_request	closede il campo merged è true	PULL_REQUEST_MERGED
pull_request	closede il campo merged è false	PULL_REQUEST_CLOSED
push	N/A	PUSH
release	rilasciato	RELEASED
release	prerelasciato	PRERELEASED
workflow_job	queued	WORKFLOW_JOB_QUEUED

 Note

Il tipo di `PULL_REQUEST_REOPENED` evento può essere utilizzato solo con GitHub Enterprise GitHub Server. Il tipo di `PRERELEASED` evento `RELEASED` può essere utilizzato GitHub solo con. Per ulteriori informazioni su `WORKFLOW_JOB_QUEUED`, consulta [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#).

Uno o più filtri opzionali

Usa un'espressione regolare per specificare un filtro. Affinché un evento attivi una build, ogni filtro all'interno del gruppo ad esso associato deve restituire `true`.

`ACTOR_ACCOUNT_ID(ACTOR_IDnella console)`

Un evento webhook attiva una build quando l'ID di un account GitHub o di GitHub Enterprise Server corrisponde al modello di espressione regolare. Questo valore si trova nella proprietà `id` dell'oggetto `sender` nel payload del webhook.

`HEAD_REF`

Un evento webhook attiva una build quando il riferimento alla testina corrisponde al modello di espressione regolare (ad esempio, `refs/heads/branch-name` `refs/tags/tag-name`). Per un evento push, il nome di riferimento si trova nella proprietà `ref` nel payload del webhook. Per gli eventi di richieste pull, il nome del ramo si trova nella proprietà `ref` dell'oggetto `head` nel payload del webhook.

`BASE_REF`

Un evento webhook attiva una build quando il riferimento di base corrisponde al modello di espressione regolare (ad esempio, `refs/heads/branch-name`). Un filtro `BASE_REF` può essere utilizzato solo con gli eventi di richiesta pull. Il nome del ramo si trova nella proprietà `ref` dell'oggetto `base` nel payload del webhook.

`FILE_PATH`

Un webhook attiva una build quando il percorso di un file modificato corrisponde al modello delle espressioni regolari. È possibile utilizzare un `FILE_PATH` filtro con eventi di richiesta GitHub push e pull ed eventi push di GitHub Enterprise Server. Non può essere utilizzato con gli eventi di pull request di GitHub Enterprise Server.

COMMIT_MESSAGE

Un webhook attiva una build quando il messaggio head commit corrisponde al modello di espressione regolare. Un COMMIT_MESSAGE filtro può essere utilizzato con gli eventi di richiesta GitHub push e pull e gli eventi push di GitHub Enterprise Server. Non può essere utilizzato con gli eventi di pull request di GitHub Enterprise Server.

TAG_NAME

Un webhook attiva una build quando il nome del tag della release corrisponde al modello di espressione regolare. È possibile utilizzare un TAG_NAME filtro con eventi di richiesta GitHub rilasciati e non rilasciati in anteprima.

RELEASE_NAME

Un webhook attiva una build quando il nome della release corrisponde al modello di espressione regolare. È possibile utilizzare un RELEASE_NAME filtro con eventi di richiesta GitHub rilasciati e non rilasciati in anteprima.

REPOSITORY_NAME

Un webhook attiva una build quando il nome del repository corrisponde al modello di espressione regolare. Un REPOSITORY_NAME filtro può essere utilizzato solo con webhook GitHub globali o organizzativi.

ORGANIZATION_NAME

Un webhook attiva una build quando il nome dell'organizzazione corrisponde al modello di espressione regolare. Un ORGANIZATION_NAME filtro può essere utilizzato solo con webhook GitHub globali.

WORKFLOW_NAME

Un webhook attiva una build quando il nome del flusso di lavoro corrisponde al modello di espressione regolare. È possibile utilizzare un WORKFLOW_NAME filtro con gli eventi di richiesta di lavoro in coda del flusso di lavoro di GitHub Actions.

Note

Puoi trovare il payload del webhook nelle impostazioni del webhook del tuo repository. GitHub

Argomenti

- [Filtra gli eventi GitHub webhook \(console\)](#)
- [Filtra gli eventi GitHub webhook \(SDK\)](#)
- [Filtra gli eventi GitHub webhook \(CloudFormation\)](#)

Filtra gli eventi GitHub webhook (console)

Utilizza le seguenti istruzioni per filtrare gli eventi GitHub webhook utilizzando. Console di gestione AWS Per ulteriori informazioni sugli eventi GitHub webhook, vedere. [GitHub eventi webhook](#)

In Primary source webhook events, seleziona quanto segue. Questa sezione è disponibile solo se hai scelto Repository nel mio GitHub account come repository di origine.

1. Al momento della creazione di un progetto, selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository).
2. Da Event type (Tipo di evento), selezionare uno o più eventi.
3. Per applicare un filtro che stabilisce quando un evento avvia una compilazione, in Start a build under these conditions (Avvia una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
4. Per applicare un filtro che stabilisce quando un evento non avvia una compilazione, in Don't start a build under these conditions (Non avviare una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
5. Scegli Aggiungi gruppo di filtri per aggiungere un altro gruppo di filtri, se necessario.

Per ulteriori informazioni, consulta le pagine [Creare un progetto di compilazione \(console\)](#) e [WebhookFilter](#) nella Documentazione di riferimento dell'API AWS CodeBuild .

In questo esempio, un gruppo di filtri webhook avvia una compilazione solo per le richieste pull:

Filter group 1

Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

► Start a build under these conditions - *optional*

► Don't start a build under these conditions - *optional*

In un esempio con due gruppi di filtri di webhook, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create, aggiornate o riaperte su ramificazioni con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/branch1$`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook avvia una compilazione per tutte le richieste pull, tranne gli eventi tag:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_REOPENED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

In questo esempio, un gruppo di filtri di webhook attiva una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^buildspec.*

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook attiva una build solo quando i file vengono modificati nelle cartelle o nelle cartelle. `src test`

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

^src/.+|^test/.+

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook attiva una build solo quando una modifica viene apportata da un utente specificato GitHub o da un utente di GitHub Enterprise Server con un ID account che corrisponde all'espressione regolare. `actor-account-id`

 Note

Per informazioni su come trovare l'ID GitHub dell'account `https://api.github.com/users/user-name`, vedi *user-name*. Dov'è il tuo nome GitHub utente.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

▼ Start a build under these conditions - optional

Add filter

Filter 2

Type

ACTOR_ACCOUNT_ID ▼

Pattern

actor-account-id

Remove

► Don't start a build under these conditions - optional

In questo esempio, un gruppo di filtri di webhook attiva una compilazione per un evento push quando il messaggio di commit HEAD corrisponde all'espressione regolare `\[CodeBuild\]`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

In questo esempio, un gruppo di filtri webhook attiva solo una build for GitHub Actions Workflow Job Events.

 Note

CodeBuild elaborerà i lavori del flusso di lavoro GitHub Actions solo se un webhook ha gruppi di filtri contenenti il filtro di eventi `WORKFLOW_JOB_QUEUED`.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

► Start a build under these conditions - optional

► Don't start a build under these conditions - optional

In questo esempio, un gruppo di filtri webhook attiva una build per un nome di workflow che corrisponde all'espressione regolare. `CI-CodeBuild`

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Filtra gli eventi GitHub webhook (SDK)

Per utilizzare l' AWS CodeBuild SDK per filtrare gli eventi webhook, utilizza il `filterGroups` campo nella sintassi di richiesta dei metodi o API. `CreateWebhook` `UpdateWebhook` Per ulteriori informazioni, consulta [WebhookFilter](#) nella documentazione di riferimento dell'API CodeBuild .

Per ulteriori informazioni sugli eventi GitHub webhook, consulta. [GitHub eventi webhook](#)

Per creare un filtro di webhook che attivi una compilazione solo per le richieste pull, inserire nella sintassi di richiesta quanto segue:

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    }  
  ]  
]
```

```
]
]
```

Per creare un filtro di webhook che attivi una compilazione solo per determinati rami, utilizzare il parametro `pattern` per specificare un'espressione regolare che filtri i nomi dei rami. In un esempio con due gruppi di filtri, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create, aggiornate o riaperte su ramificazioni con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/myBranch$`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

Il parametro `excludeMatchedPattern` consente di indicare quali eventi non attivano una compilazione. In questo esempio, viene attivata una compilazione per tutte le richieste, tranne gli eventi `tag`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

È possibile creare un filtro che attivi una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare nell'argomento `pattern`. In questo esempio, il gruppo di filtri indica che una compilazione si attiva solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^buildspec.*"  
    }  
  ]  
]
```

In questo esempio, il gruppo di filtri specifica che una build viene attivata solo quando i file vengono modificati nelle cartelle o nelle `src` cartelle. `test`

```
"filterGroups": [  
  [  
    {  
      "type": "FILE_PATH",  
      "pattern": "src/test/*"  
    }  
  ]  
]
```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "FILE_PATH",
    "pattern": "^src/.+|^test/.+"
  }
]
```

È possibile creare un filtro che attiva una compilazione solo quando una modifica viene apportata da un utente specificato GitHub o da un utente di GitHub Enterprise Server con ID account. `actor-account-id`

Note

Per informazioni su come trovare l'ID GitHub dell'account <https://api.github.com/users/user-name>, vedi *user-name*. Dov'è il tuo nome GitHub utente.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

È possibile creare un filtro che attiva una compilazione solo quando il messaggio di commit HEAD corrisponde all'espressione regolare nell'argomento del modello. In questo esempio, il gruppo di filtri specifica che viene attivata una compilazione solo quando il messaggio di commit HEAD dell'evento push corrisponde all'espressione regolare `\[CodeBuild\]`.


```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\\[CodeBuild\\]"  
    }  
  ]  
]
```

Per creare un filtro webhook che attivi solo i job del flusso di lavoro Build for GitHub Actions, inserisci quanto segue nella sintassi della richiesta:

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "WORKFLOW_JOB_QUEUED"  
    }  
  ]  
]
```

Filtra gli eventi GitHub webhook ()CloudFormation

Per utilizzare un CloudFormation modello per filtrare gli eventi del webhook, utilizzate la proprietà del AWS CodeBuild `FilterGroups` progetto.

Per ulteriori informazioni sugli eventi GitHub webhook, vedere. [GitHub eventi webhook](#)

La parte di un modello CloudFormation in formato YAML riportata di seguito crea due gruppi di filtri. Insieme, questi attivano una compilazione quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica che le richieste pull vengono create o aggiornate sui rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` da un GitHub utente che non dispone di un ID 12345 account.
- Il secondo gruppo di filtri specifica le richieste push create su file con nomi che corrispondono all'espressione regolare `README` in rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/.*`.

- Il terzo gruppo di filtri specifica una richiesta push con un messaggio di commit HEAD corrispondente all'espressione regolare `\[CodeBuild\]`.
- Il quarto gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions con un nome di flusso di lavoro corrispondente all'espressione regolare. `\[CI-CodeBuild\]`

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
        - Type: ACTOR_ACCOUNT_ID
          Pattern: 12345
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: HEAD_REF
          Pattern: ^refs/heads/. *
        - Type: FILE_PATH
          Pattern: READ_ME
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: COMMIT_MESSAGE
          Pattern: \[CodeBuild\]
        - Type: FILE_PATH
```

```
Pattern: ^src/.+|^test/.+
- - Type: EVENT
Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]
```

GitLab webhook di gruppo

È possibile utilizzare i webhook di CodeBuild GitLab gruppo per avviare build su eventi webhook da qualsiasi repository all'interno di un gruppo. GitLab I webhook di gruppo funzionano con tutti i tipi di eventi GitLab webhook esistenti e possono essere configurati aggiungendo una configurazione di ambito durante la creazione di un webhook. CodeBuild Puoi anche utilizzare i webhook di gruppo per [configurare GitLab runner ospitati autonomamente al fine di ricevere WORKFLOW_JOB_QUEUED eventi da più repository CodeBuild all'interno](#) di un singolo progetto.

Argomenti

- [Configura un webhook di gruppo GitLab](#)
- [Filtra gli eventi webhook GitLab di gruppo \(console\)](#)
- [Filtra gli eventi webhook di GitLab gruppo \(\)CloudFormation](#)

Configura un webhook di gruppo GitLab

I passaggi principali per configurare un GitLab webhook di gruppo sono i seguenti. Per ulteriori informazioni sui GitLab webhook di gruppo, vedere. [GitLab webhook di gruppo](#)

1. Imposta la posizione di origine del progetto su.
CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
2. Nella configurazione dell'ambito del webhook, imposta l'ambito su. GITLAB_GROUP
3. Specificate un nome come parte della configurazione dell'ambito del webhook. Per i webhook di gruppo, questo è il nome del gruppo.

Note

Se il tipo di sorgente del progetto è GITLAB_SELF_MANAGED, dovrai anche specificare un dominio come parte della configurazione dell'ambito del webhook.

4. (Facoltativo) Se desideri ricevere solo eventi webhook per repository specifici all'interno della tua organizzazione o azienda, puoi specificarli `REPOSITORY_NAME` come filtro durante la creazione del webhook.
5. Quando crei un webhook di gruppo, assicurati di disporre delle autorizzazioni necessarie per creare webhook a livello di gruppo all'interno. CodeBuild GitLab Per farlo, puoi usare però. CodeBuild OAuth CodeConnections Per ulteriori informazioni, consulta [GitLab accesso in CodeBuild](#).

Tieni presente che i webhook di gruppo funzionano con qualsiasi tipo di evento GitLab webhook esistente.

Filtra gli eventi webhook GitLab di gruppo (console)

Quando crei un GitLab progetto tramite la console, seleziona le seguenti opzioni per creare un webhook di GitLab gruppo all'interno del progetto. Per ulteriori informazioni sui GitLab webhook di gruppo, consulta. [GitLab webhook di gruppo](#)

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
 - In Source (Origine):
 - Come fornitore di origine, scegli Self Managed GitLab. GitLab
 - Per Repository, scegli un webhook GitLabcon ambito.

Il GitLab repository verrà impostato automaticamente su `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, che è la posizione di origine richiesta per i webhook di gruppo.

Note

Quando utilizzi webhook di gruppo, assicurati di disporre delle autorizzazioni necessarie per creare webhook a CodeBuild livello di gruppo all'interno. GitLab Se utilizzi una [OAuthconnessione esistente](#), potrebbe essere necessario rigenerare la connessione per concedere questa autorizzazione. CodeBuild

Source

Add source

Source 1 - Primary

Source provider

GitLab

Credential

☒ Default source credential
Use your account's default source credential to apply to all projects

☐ Custom source credential
Use a custom source credential to override your account's default settings

Successfully connected through CodeConnections - [open resource](#)

Manage default source credential

Repository

☐ Repository in my GitLab account

☒ GitLab scoped webhook

Repository

CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION

- Negli eventi webhook della fonte primaria:
- Per Nome gruppo, inserisci il nome del gruppo.

Se il tipo di sorgente del progetto è `GITLAB_SELF_MANAGED`, devi anche specificare un dominio come parte della configurazione del gruppo webhook. Ad esempio, se l'URL del gruppo è `https://domain.com/group/group-name`, allora il dominio è `https://domain.com`.

Note

Questo nome non può essere modificato dopo la creazione del webhook. Per modificare il nome, puoi eliminare e ricreare il webhook. Se desideri rimuovere completamente il webhook, puoi anche aggiornare la posizione di origine del progetto in un repository. GitLab

Primary source webhook events [Info](#)[Add filter group](#)Webhook - *optional* [Info](#) ☒ Rebuild every time a code change is pushed to this repository

Group name

Your GitLab group name.

Build type

☒ **Single build**
Triggers single build☐ **Batch build**
Triggers multiple builds as single execution[► Additional configuration](#)

- (Facoltativo) Nei gruppi di filtri degli eventi Webhook, puoi specificare quali [eventi desideri attivare una](#) nuova build. Puoi anche specificare `REPOSITORY_NAME` come filtro di attivare solo le build sugli eventi webhook provenienti da repository specifici.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1[Remove filter group](#)Event type - *optional*

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED [▼ Start a build under these conditions - optional](#)[Add filter](#)**Filter 1**

Type

Pattern

[Remove](#)

Puoi anche impostare il tipo di evento su per configurare i corridori ospitati WORKFLOW_JOB_QUEUED autonomamente. GitLab Per ulteriori informazioni, consulta [Runner GitLab autogestiti in AWS CodeBuild](#).

3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.

Filtra gli eventi webhook di GitLab gruppo ()CloudFormation

Per utilizzare un CloudFormation modello per filtrare gli eventi webhook di gruppo, utilizzate la proprietà del AWS CodeBuild ScopeConfiguration progetto. Per ulteriori informazioni sui GitLab webhook di gruppo, vedere. [GitLab webhook di gruppo](#)

La seguente parte di un modello in formato YAML crea quattro gruppi di CloudFormation filtri. Insieme, attivano una build quando uno o tutti restituiscono true:

- Il primo gruppo di filtri specifica che le richieste pull vengono create o aggiornate sui rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` da un GitLab utente che non dispone di un ID 12345 account.
- Il secondo gruppo di filtri specifica le richieste push create su file con nomi che corrispondono all'espressione regolare `READ_ME` in rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/.*`.
- Il terzo gruppo di filtri specifica una richiesta push con un messaggio di commit `HEAD` corrispondente all'espressione regolare `\[CodeBuild\]`.
- Il quarto gruppo di filtri specifica un nome di GitLab CI/CD pipeline job request with a CI/CD pipeline corrispondente all'espressione regolare. `\[CI-CodeBuild\]`

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
```

```
Type: GITLAB
Location: source-location
Triggers:
  Webhook: true
  ScopeConfiguration:
    Name: group-name
    Scope: GITLAB_GROUP
  FilterGroups:
    - - Type: EVENT
      Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
    - Type: BASE_REF
      Pattern: ^refs/heads/main$
      ExcludeMatchedPattern: false
    - Type: ACTOR_ACCOUNT_ID
      Pattern: 12345
      ExcludeMatchedPattern: true
    - - Type: EVENT
      Pattern: PUSH
    - Type: HEAD_REF
      Pattern: ^refs/heads/.+
    - Type: FILE_PATH
      Pattern: README
      ExcludeMatchedPattern: true
    - - Type: EVENT
      Pattern: PUSH
    - Type: COMMIT_MESSAGE
      Pattern: \[CodeBuild\]
    - Type: FILE_PATH
      Pattern: ^src/.+|^test/.+
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
    - Type: WORKFLOW_NAME
      Pattern: \[CI-CodeBuild\]
```

GitLab webhook manuali

È possibile configurare i GitLab webhook manuali per CodeBuild impedire il tentativo automatico di creare un webhook al loro interno. GitLab CodeBuild restituisce un URL di payload come parte della chiamata per creare il webhook e può essere utilizzato per creare manualmente il webhook al suo interno. GitLab Anche se la creazione di un webhook nel tuo GitLab account non CodeBuild è consentita, puoi comunque creare manualmente un webhook per il tuo progetto di build.

Usa la seguente procedura per creare un GitLab webhook manuale.

Per creare un webhook GitLab manuale

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
 - In Source (Origine):
 - Per Source provider, scegli. GitLab
 - Per Repository, scegli Repository nel mio GitLab account.
 - Nel campo Repository URL (URL repository), inserire **https://gitlab.com/*user-name*/*repository-name***
 - Negli eventi webhook di Primary Source:
 - Per Webhook: facoltativo, scegli Ricostruisci ogni volta che viene inviata una modifica al codice in questo repository.
 - Scegli Configurazione aggiuntiva e per Creazione manuale: facoltativo, scegli Crea manualmente un webhook per questo repository nella console. GitLab .
3. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione. Prendi nota dei valori Payload URL e Secret poiché li utilizzerai in seguito.
4. Apri la GitLab console all'indirizzo [https://gitlab.com/*user-name*/*repository-name*/-/hooks](https://gitlab.com/<i>user-name</i>/<i>repository-name</i>/-/hooks) e scegli Aggiungi nuovo webhook.
 - Per URL, inserisci il valore dell'URL di payload che hai preso nota in precedenza.
 - Per Secret token, inserisci il valore Secret di cui hai preso nota in precedenza.
 - Configura i singoli eventi a cui verrà inviato un payload del webhook. CodeBuild Per Trigger, scegli tra i seguenti eventi: eventi Push, eventi Merge request, eventi Releases ed eventi Job. Per ulteriori informazioni sui tipi di eventi supportati da CodeBuild, consulta [GitLab eventi webhook](#).
5. Scegli Aggiungi webhook.

GitLab eventi webhook

È possibile utilizzare i gruppi di filtri webhook per specificare quali eventi GitLab webhook attivano una build. Ad esempio, è possibile specificare che una build venga attivata solo per le modifiche a rami specifici.

È possibile creare uno o più gruppi di filtri webhook per specificare quali eventi webhook attivano una build. Una build viene attivata se un gruppo di filtri restituisce true, il che si verifica quando tutti i filtri del gruppo restituiscono true. Quando si crea un gruppo di filtri, si specifica:

Un evento

Infatti GitLab, puoi scegliere uno o più dei seguenti

eventi: PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED, PULL_REQUEST_REOPENED, PULL_REQUEST_CLOSED, RELEASED, e WORKFLOW_JOB_QUEUED.

Il tipo di evento del webhook è nell'intestazione del campo X-GitLab-Event. La tabella seguente mostra il modo in cui i valori dell'intestazione X-GitLab-Event sono mappati ai tipi di eventi. Per l'evento Merge Request Hook webhook, il payload `object_attributes.action` conterrà informazioni aggiuntive sul tipo di richiesta di unione.

Valore intestazione X-GitLab-Event	<code>object_attributes.action</code>	Tipo di evento
Push Hook	N/D	PUSH
Merge Request Hook	aperto	PULL_REQUEST_CREATED
Merge Request Hook	aggiorna	PULL_REQUEST_UPDATED
Merge Request Hook	merge	PULL_REQUEST_MERGED
Merge Request Hook	riapertura	PULL_REQUEST_REOPENED
Merge Request Hook	close	PULL_REQUEST_CLOSED
Release Hook	creare, aggiornare	RELEASED
Job Hook	N/D	WORKFLOW_JOB_QUEUED

Infatti PULL_REQUEST_MERGED, se una pull request viene unita alla strategia squash e il ramo pull request viene chiuso, il commit di pull request originale non esiste più. In questo caso, la variabile di `CODEBUILD_WEBHOOK_MERGE_COMMIT` ambiente contiene l'identificatore dello squashed merge commit.

Uno o più filtri opzionali

Usa un'espressione regolare per specificare un filtro. Affinché un evento attivi una build, ogni filtro all'interno del gruppo ad esso associato deve restituire true.

`ACTOR_ACCOUNT_ID(ACTOR_IDnella console)`

Un evento webhook attiva una build quando l'ID di un GitLab account corrisponde al modello di espressione regolare. Questo valore è presente nella proprietà `account_id` dell'oggetto `actor` nel payload del filtro webhook.

`HEAD_REF`

Un evento webhook attiva una build quando il riferimento alla testina corrisponde al modello di espressione regolare (ad esempio, `and`). `refs/heads/branch-name` `refs/tags/tag-name` Un filtro `HEAD_REF` valuta il nome di riferimento Git per il ramo o il tag. Il nome del ramo o del tag è presente nel campo `name` dell'oggetto `new` nell'oggetto `push` del payload del webhook. Per gli eventi di richiesta `pull`, il nome del ramo è presente nel campo `name` dell'oggetto `branch` dell'oggetto `source` nel payload del webhook.

`BASE_REF`

Un evento webhook attiva una build quando il riferimento di base corrisponde al modello di espressione regolare. Un filtro `BASE_REF` funziona solo con gli eventi di richiesta `pull` (ad esempio, `refs/heads/branch-name`). Un filtro `BASE_REF` valuta il nome di riferimento Git per il ramo. Il nome del ramo è presente nel campo `name` dell'oggetto `branch` nell'oggetto `destination` nel payload del webhook.

`FILE_PATH`

Un webhook attiva una build quando il percorso di un file modificato corrisponde al modello di espressione regolare.

`COMMIT_MESSAGE`

Un webhook attiva una build quando il messaggio di head commit corrisponde al modello di espressione regolare.

`WORKFLOW_NAME`

Un webhook attiva una build quando il nome del flusso di lavoro corrisponde al modello di espressione regolare.

 Note

Puoi trovare il payload del webhook nelle impostazioni del webhook del tuo repository. [GitLab](#)

Argomenti

- [Filtra gli eventi GitLab webhook \(console\)](#)
- [Filtra gli eventi GitLab webhook \(SDK\)](#)
- [Filtra gli GitLab eventi webhook \(\)CloudFormation](#)

Filtra gli eventi GitLab webhook (console)

Utilizza le seguenti istruzioni per utilizzare per Console di gestione AWS filtrare gli eventi webhook. Per ulteriori informazioni sugli eventi GitLab webhook, vedere. [GitLab eventi webhook](#)

1. Al momento della creazione di un progetto, selezionare Rebuild every time a code change is pushed to this repository (Ricompila ogni volta che viene inviata una modifica del codice a questo repository).
2. Da Event type (Tipo di evento), selezionare uno o più eventi.
3. Per applicare un filtro che stabilisce quando un evento avvia una compilazione, in Start a build under these conditions (Avvia una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
4. Per applicare un filtro che stabilisce quando un evento non avvia una compilazione, in Don't start a build under these conditions (Non avviare una compilazione in queste condizioni) aggiungere uno o più filtri facoltativi.
5. Per aggiungere un altro gruppo di filtri, scegliere Add filter group (Aggiungi gruppo di filtri).

Per ulteriori informazioni, consulta le pagine [Creare un progetto di compilazione \(console\)](#) e [WebhookFilter](#) nella Documentazione di riferimento dell'API AWS CodeBuild .

In questo esempio, un gruppo di filtri webhook avvia una compilazione solo per le richieste pull:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕PULL_REQUEST_UPDATED ✕PULL_REQUEST_MERGED ✕

► **Start a build under these conditions - *optional***

► **Don't start a build under these conditions - *optional***

In un esempio con due gruppi di filtri, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create o aggiornate su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/branch1!`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/branch1$`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ×PULL_REQUEST_UPDATED ×

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Filter 2

Type

Pattern

[Remove](#)

► **Don't start a build under these conditions - optional**

Filter group 2

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ×

Filter 1

Type

In questo esempio, un gruppo di filtri webhook avvia una compilazione per tutte le richieste pull, tranne gli eventi tag:

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_MERGED](#) ✕

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

In questo esempio, un gruppo di filtri di webhook attiva una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

In questo esempio, un gruppo di filtri webhook attiva una build solo quando i file vengono modificati nelle cartelle o nelle cartelle. `src test`

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

FILE_PATH ▼

Pattern

`^src/.+|^test/.+`

Remove

► Don't start a build under these conditions - optional

In questo esempio, un gruppo di filtri webhook attiva una build solo quando viene apportata una modifica da un GitLab utente che non dispone di un ID account corrispondente all'espressione regolare. `actor-account-id`

Note

Per informazioni su come trovare l'ID GitLab del tuo account <https://api.github.com/users/user-name>, vedi `user-name` Dov'è il tuo nome GitLab utente.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

In questo esempio, un gruppo di filtri di webhook attiva una compilazione per un evento push quando il messaggio di commit HEAD corrisponde all'espressione regolare `\[CodeBuild\]`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) [×](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Filtra gli eventi GitLab webhook (SDK)

Per utilizzare l' AWS CodeBuild SDK per filtrare gli eventi webhook, utilizza il `filterGroups` campo nella sintassi di richiesta dei metodi o API. `CreateWebhook` `UpdateWebhook` Per ulteriori informazioni, consulta [WebhookFilter](#) nella documentazione di riferimento dell'API CodeBuild .

Per ulteriori informazioni sugli eventi GitLab webhook, consulta. [GitLab eventi webhook](#)

Per creare un filtro di webhook che attivi una compilazione solo per le richieste pull, inserire nella sintassi di richiesta quanto segue:

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```

```
    "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED"
  }
]
]
```

Per creare un filtro di webhook che attivi una compilazione solo per determinati rami, utilizzare il parametro `pattern` per specificare un'espressione regolare che filtri i nomi dei rami. In un esempio con due gruppi di filtri, una compilazione si attiva quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica le richieste pull create o aggiornate su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` e riferimenti head che corrispondono a `^refs/heads/myBranch$`.
- Il secondo gruppo di filtri specifica le richieste push su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

```
]
```

Il parametro `excludeMatchedPattern` consente di indicare quali eventi non attivano una compilazione. In questo esempio, viene attivata una compilazione per tutte le richieste, tranne gli eventi `tag`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Puoi creare un filtro che attiva una build solo quando una modifica viene apportata da un GitLab utente con ID account. `actor-account-id`

Note

Per informazioni su come trovare l'ID GitLab del tuo account <https://api.github.com/users/user-name>, vedi *user-name*. Dov'è il tuo nome GitLab utente.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",  
      "pattern": "actor-account-id"  
    }  
  ]  
]
```

```
]
]
```

È possibile creare un filtro che attivi una compilazione solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare nell'argomento `pattern`. In questo esempio, il gruppo di filtri indica che una compilazione si attiva solo in caso di modifica dei file con nomi che corrispondono all'espressione regolare `^buildspec.*`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

In questo esempio, il gruppo di filtri specifica che una build viene attivata solo quando i file vengono modificati nelle cartelle `src` o `test` nelle cartelle.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

È possibile creare un filtro che attiva una compilazione solo quando il messaggio di commit HEAD corrisponde all'espressione regolare nell'argomento del modello. In questo esempio, il gruppo di filtri specifica che viene attivata una compilazione solo quando il messaggio di commit HEAD dell'evento push corrisponde all'espressione regolare `\[CodeBuild\]`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "COMMIT_MESSAGE",  
      "pattern": "\\[CodeBuild\\]"  
    }  
  ]  
]
```

Filtra gli GitLab eventi webhook ()CloudFormation

Per utilizzare un CloudFormation modello per filtrare gli eventi del webhook, utilizzate la proprietà del AWS CodeBuild `FilterGroups` progetto. Per ulteriori informazioni sugli eventi GitLab webhook, vedere. [GitLab eventi webhook](#)

La parte di un modello CloudFormation in formato YAML riportata di seguito crea due gruppi di filtri. Insieme, questi attivano una compilazione quando uno o entrambi risultano soddisfatti:

- Il primo gruppo di filtri specifica che le richieste pull vengono create o aggiornate sui rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/main$` da un GitLab utente che non dispone di un ID 12345 account.
- Il secondo gruppo di filtri specifica le richieste push create su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/.*`.
- Il terzo gruppo di filtri specifica una richiesta push con un messaggio di commit HEAD corrispondente all'espressione regolare `\\[CodeBuild\\]`.
- Il quarto gruppo di filtri specifica una richiesta di lavoro del flusso di lavoro GitHub Actions con un nome di flusso di lavoro corrispondente all'espressione regolare. `\\[CI-CodeBuild\\]`

```
CodeBuildProject:  
  Type: AWS::CodeBuild::Project  
  Properties:  
    Name: MyProject  
    ServiceRole: service-role  
    Artifacts:
```

```
Type: NO_ARTIFACTS
Environment:
  Type: LINUX_CONTAINER
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:5.0
Source:
  Type: GITLAB
  Location: source-location
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
    - - Type: BASE_REF
      Pattern: ^refs/heads/main$
      ExcludeMatchedPattern: false
    - - Type: ACTOR_ACCOUNT_ID
      Pattern: 12345
      ExcludeMatchedPattern: true
    - - Type: EVENT
      Pattern: PUSH
    - - Type: HEAD_REF
      Pattern: ^refs/heads/. *
    - - Type: EVENT
      Pattern: PUSH
    - - Type: COMMIT_MESSAGE
      Pattern: \[CodeBuild\]
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
    - - Type: WORKFLOW_NAME
      Pattern: \[CI-CodeBuild\]
```

Webhook manuali Buildkite

Attualmente, CodeBuild richiede che tutti i webhook Buildkite vengano creati manualmente.

CodeBuild restituisce un URL di payload come parte della chiamata per creare il webhook, che può essere utilizzato per creare manualmente il webhook all'interno di Buildkite.

Usa la seguente procedura per creare un webhook manuale Buildkite.

Per creare un CodeBuild progetto con un webhook

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Creare un progetto di compilazione. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
3. Nella configurazione del progetto, scegli Runner project.

In Runner:

- Per il provider Runner, scegli Buildkite.
 - Per il token dell'agente Buildkite, scegli Crea un nuovo token agente utilizzando la pagina di creazione segreta. Ti verrà richiesto di creare un nuovo segreto in AWS Secrets Manager con un valore segreto uguale al token dell'agente Buildkite che hai generato sopra.
 - (Facoltativo) Se desideri utilizzare credenziali CodeBuild gestite per il tuo lavoro, seleziona il provider del repository di origine del tuo lavoro nelle opzioni delle credenziali di origine di Buildkite e verifica che le credenziali siano configurate per il tuo account. Inoltre, verifica che la tua pipeline Buildkite utilizzi Checkout utilizzando HTTPS.
4. • In Environment (Ambiente):
 - Scegli un'immagine Environment supportata e Calcola. Tieni presente che hai la possibilità di sovrascrivere le impostazioni dell'immagine e dell'istanza utilizzando un'etichetta nel flusso di lavoro GitHub Actions (YAML). Per ulteriori informazioni, consulta [Passaggio 2: aggiorna il flusso di lavoro di GitHub Actions \(YAML\)](#)
 - In Buildspec:
 - Nota che il tuo buildspec verrà ignorato a meno che non venga aggiunto come etichetta. `buildspec-override:true` Invece, lo CodeBuild sovrascriverà per utilizzare i comandi che configureranno il runner self-hosted.
 5. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.
 6. Salva l'URL del payload e i valori segreti dal popup Create Webhook. Segui le istruzioni nel popup per creare un nuovo webhook dell'organizzazione Buildkite.

Pull richiedi l'approvazione dei commenti

CodeBuild supporta le politiche di compilazione delle pull request che forniscono un controllo aggiuntivo sulle build attivate dalle richieste pull. Potresti non voler creare automaticamente richieste pull da utenti sconosciuti fino a quando le loro modifiche non saranno riviste. Questa funzionalità ti consente di richiedere a uno dei membri del tuo team di rivedere prima il codice e poi eseguire la pipeline. Questa è comunemente usata come misura di sicurezza quando si crea un codice inviato da collaboratori sconosciuti.

Le politiche di compilazione delle pull request consentono di controllare quando vengono generati i CodeBuild trigger per le richieste pull in base alle autorizzazioni e allo stato di approvazione del contribuente. Ciò è particolarmente importante per gli archivi pubblici o gli archivi che accettano contributi da collaboratori esterni.

Se abilitata, questa funzionalità assicura che le build vengano attivate solo per le richieste pull quando:

- La pull request viene creata da un collaboratore affidabile.
- Un collaboratore affidabile approva la pull request pubblicando un commento specifico.

Come funziona

Collaboratori fidati

Un collaboratore affidabile è un utente il cui ruolo attuale nel sistema di controllo del codice sorgente è impostato nella politica basata sulle pull request come ruoli di approvatore. Quando un collaboratore affidabile crea una pull request, CodeBuild attiva automaticamente la build, mantenendo il comportamento corrente.

Collaboratori non attendibili

Un collaboratore non attendibile è un utente il cui ruolo non è impostato nell'elenco dei ruoli di approvatore. Quando un contribuente non attendibile crea una pull request:

1. CodeBuild contrassegna lo stato di compilazione come «Failed» con il messaggio «Pull request approval required for start a build».
2. Un collaboratore affidabile deve esaminare le modifiche e pubblicare un commento / `codebuild_run(<SHA_OF_THE_LATEST_COMMIT>)` per attivare la build. Ad esempio, / `codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`.
3. CodeBuild convalida le autorizzazioni del commentatore e attiva la build se approvata.
4. I risultati della build vengono riportati nella pagina di pull request.

Sintassi di approvazione dei commenti

I collaboratori attendibili possono approvare le build utilizzando i seguenti formati di commento:

- `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`- I trigger si basano sul commit SHA specificato.

Configurazione

Comportamento predefinito

La politica di compilazione delle pull request è abilitata per impostazione predefinita per tutti i progetti appena creati CodeBuild .

Parametri dell'API

La politica di compilazione della pull request è configurata utilizzando il `PullRequestBuildPolicy` parametro nelle seguenti azioni:

- `CreateWebhook`
- `UpdateWebhook`

`PullRequestBuildPolicy` struttura

```
{
  "requiresCommentApproval": "string",
  "approverRoles": ["string", ...]
}
```

`requiresCommentApproval`

Specifica quando è richiesta l'approvazione basata sui commenti prima di attivare una build on pull request. Questa impostazione determina se le build vengono eseguite automaticamente o richiedono l'approvazione esplicita tramite commenti.

Tipo: String

Valori validi:

- `DISABLED`- Le build si attivano automaticamente senza richiedere l'approvazione dei commenti.
- `FORK_PULL_REQUESTS`- Solo le richieste pull provenienti da repository biforcati richiedono l'approvazione dei commenti (a meno che il collaboratore non sia uno dei ruoli di approvazione).
- `ALL_PULL_REQUESTS`- Tutte le richieste pull richiedono l'approvazione dei commenti prima dell'esecuzione delle build (a meno che il collaboratore non sia uno dei ruoli di approvazione). Si tratta del valore di default.

`approverRoles`

Elenco dei ruoli del repository che dispongono dei privilegi di approvazione per le build di pull request quando è richiesta l'approvazione dei commenti. Solo gli utenti con questi ruoli possono

fornire approvazioni di commenti valide. Se un collaboratore di pull request è uno di questi ruoli, le relative build di pull request verranno attivate automaticamente.

Tipo: array di stringhe

Valori validi per i GitHub progetti (i valori sono mappati ai ruoli): GitHub

- GITHUB_ADMIN- Amministratori del repository
- GITHUB_MAINTAIN- Manutentori del repository
- GITHUB_WRITE- Utente con autorizzazioni di scrittura
- GITHUB_TRIAGE- Utente con autorizzazioni di triage
- GITHUB_READ- Utente con autorizzazioni di lettura
- Impostazione predefinita: ["GITHUB_ADMIN", "GITHUB_MAINTAINER", "GITHUB_WRITE"]

Valori validi per i GitLab progetti (i valori sono mappati ai GitLab ruoli):

- GITLAB_OWNER- Proprietario del repository
- GITLAB_MAINTAINER- Gestore del repository
- GITLAB_DEVELOPER- Utente con autorizzazioni di sviluppatore
- GITLAB_REPORTER- Utente con autorizzazioni di reporter
- GITLAB_PLANNER- Utente con autorizzazioni di pianificazione
- GITLAB_GUEST - Utente con autorizzazioni da ospite
- Impostazione predefinita: ["GITLAB_OWNER", "GITLAB_MAINTAINER", "GITLAB_DEVELOPER"]

Valori validi per i progetti Bitbucket (i valori sono mappati ai ruoli di Bitbucket):

- BITBUCKET_ADMIN - Amministratore del repository
- BITBUCKET_WRITE- Utente con autorizzazioni di scrittura
- BITBUCKET_READ - Utente con autorizzazioni di lettura
- Impostazione predefinita: ["BITBUCKET_ADMIN", "BITBUCKET_WRITE"]

Esempi

Abilita l'approvazione dei commenti per tutte le richieste pull

Per utilizzare l' AWS CodeBuild SDK per abilitare o disabilitare la politica Pull Request Build per un webhook, utilizza il `pullRequestBuildPolicy` campo nella sintassi della richiesta dei metodi `CreateWebhook` o `UpdateWebhook` API. Per ulteriori informazioni, consulta [WebhookFilter](#) nella documentazione di riferimento dell'API CodeBuild .

Gli utenti con ruoli Github Admin, Maintain e Write verranno trattati come collaboratori affidabili.

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "ALL_PULL_REQUESTS",
  "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]
}
```

Abilita l'approvazione dei commenti solo per gli amministratori e i manutentori del repository

Gli utenti con GitHub i ruoli Admin, Maintain verranno trattati come collaboratori affidabili.

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "FORK_PULL_REQUESTS",
  "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAINER"]
}
```

Disabilita l'approvazione dei commenti

```
"pullRequestBuildPolicy": {
  "requiresCommentApproval": "DISABLED"
}
```

AWS CloudFormation

Per utilizzare un AWS CloudFormation modello per abilitare o disabilitare la politica Pull Request Build per un webhook, utilizzate la `PullRequestBuildPolicy` proprietà. La seguente parte di un AWS CloudFormation modello in formato YAML crea un progetto con un webhook che ha la Pull Request Build Policy abilitata per tutte le richieste pull. Mantieni e amministra i ruoli specificati come approvatori.

```
CodeBuildProject:
```

```
Type: AWS::CodeBuild::Project
Properties:
  Name: MyProject
  ServiceRole: service-role
  Artifacts:
    Type: NO_ARTIFACTS
  Environment:
    Type: LINUX_CONTAINER
    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/standard:5.0
  Source:
    Type: BITBUCKET
    Location: source-location
  Triggers:
    Webhook: true
    FilterGroups:
      - - Type: EVENT
        Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
      - Type: BASE_REF
        Pattern: ^refs/heads/main$
        ExcludeMatchedPattern: false
    PullRequestBuildPolicy:
      RequiresCommentApproval: ALL_PULL_REQUESTS
      ApproverRoles:
        - GITHUB_MAINTAIN
        - GITHUB_ADMIN
```

Configurazione della console

Per utilizzare la console di AWS gestione per filtrare gli eventi dei webhook:

1. Per l'approvazione dei commenti, seleziona disabilitato o abilitato per tutte le richieste pull (ALL_PULL_REQUEST) o solo per le richieste pull da forks (FORK_PULL_REQUEST).
2. Per i ruoli di approvatore, seleziona i ruoli del repository che dispongono dei privilegi di approvazione per le build di pull request quando è richiesta l'approvazione dei commenti.

Per ulteriori informazioni, consulta le pagine [Creare un progetto di compilazione \(console\)](#) e [WebhookFilter](#) nella Documentazione di riferimento dell'API CodeBuild .

▼ **Primary source webhook events** [Info](#)

Webhook - optional [Info](#)

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

Build type

☒ **Single build**
Triggers single build

☐ **Batch build**
Triggers multiple builds as single execution

Comment approval

ALL_PULL_REQUESTS ▼

Approver roles

BITBUCKET_WRITE ✕

BITBUCKET_ADMIN ✕

► **Webhook event filter groups**

Add filter group

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Visualizza i dettagli di un progetto di costruzione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o visualizzare AWS SDKs i dettagli di un progetto di compilazione in CodeBuild. AWS CLI

Argomenti

- [Visualizzazione dei dettagli di un progetto di compilazione \(console\)](#)
- [Visualizzazione dei dettagli di un progetto di compilazione \(AWS CLI\)](#)
- [Visualizzazione dei dettagli di un progetto di compilazione \(AWS SDKs\)](#)

Visualizzazione dei dettagli di un progetto di compilazione (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).

 Note

Per impostazione predefinita, vengono visualizzati solo i 10 progetti di compilazione più recenti. Per visualizzare altri progetti di compilazione, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

3. Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), selezionare il collegamento del progetto di compilazione.
4. Nella **project-name** pagina Build project:, scegli Build details.

Visualizzazione dei dettagli di un progetto di compilazione (AWS CLI)

Eseguire il comando batch-get-projects:

```
aws codebuild batch-get-projects --names names
```

Nel comando precedente, sostituire il seguente segnaposto:

- **names**: Stringa obbligatoria utilizzata per indicare uno o più nomi di progetti di build di cui visualizzare i dettagli. Per specificare più di un progetto di compilazione, separare ogni nome di progetto con uno spazio. È possibile specificare fino a 100 nomi di progetti di compilazione. Per ottenere un elenco di progetti di compilazione, consulta [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#).

Ad esempio, se si esegue questo comando:

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2  
my-other-demo-project
```

Nell'output potrebbe essere visualizzato un risultato come il seguente. Per brevità, le ellissi (...) vengono utilizzate per rappresentare i dati omessi.

```
{  
  "projectsNotFound": [  

```



```
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": codebuild-demo-project,
      ...
    },
    {
      ...
      "name": codebuild-demo-project2",
      ...
    }
  ]
}
```

Nell'output precedente, l'array `projectsNotFound` elenca qualsiasi nome di progetto di compilazione che è stato specificato, ma non trovato. L'array `projects` elenca i dettagli per ogni progetto di compilazione di cui sono state trovate informazioni. Per brevità, i dettagli del progetto di compilazione sono stati omessi dall'output precedente. Per maggiori informazioni, consultare l'output di [Creazione di un progetto di compilazione \(AWS CLI\)](#).

Il `batch-get-projects` comando non supporta il filtraggio per determinati valori di proprietà, ma è possibile scrivere uno script che enumeri le proprietà di un progetto. Ad esempio, il seguente script di shell Linux enumera i progetti nell'area corrente per l'account corrente e stampa l'immagine utilizzata da ogni progetto.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.

imageName=""

function getImageName(){
  local environmentValues=(${1//$'\t' / })
  imageName=${environmentValues[1]}
}

function processProjectInfo() {
  local projectInfo=$1
```

```
while IFS=$'\t' read -r section value; do
    if [[ "$section" == *"ENVIRONMENT"* ]]; then
        getImageName "$value"
    fi
done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
    if [[ "$projectName" != *"PROJECTS"* ]]; then
        echo "=====

        # Get the detailed information for the project.
        projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

        processProjectInfo "$projectInfo"

        printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
    fi
done
```

Per ulteriori informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta. [Guida di riferimento alla riga di comando](#)

Visualizzazione dei dettagli di un progetto di compilazione (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, vedere [AWS SDKs e riferimento agli strumenti](#).

Visualizza i nomi dei progetti di compilazione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs visualizzare un elenco di progetti di compilazione in CodeBuild. AWS CLI

Argomenti

- [Visualizzazione di un elenco di nomi di progetti di compilazione \(console\)](#)
- [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#)

- [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS SDKs\)](#)

Visualizzazione di un elenco di nomi di progetti di compilazione (console)

È possibile visualizzare un elenco di progetti di compilazione in una AWS regione nella console. Le informazioni includono il nome, il provider di origine, il repository, lo stato di compilazione più recente e la descrizione, se presente.

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).

Note

Per impostazione predefinita, vengono visualizzati solo i 10 progetti di compilazione più recenti. Per visualizzare altri progetti di compilazione, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Visualizzazione di un elenco di nomi di progetti di compilazione (AWS CLI)

Eseguire il comando list-projects:

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

Nel comando precedente, sostituire i seguenti segnaposto:

- *sort-by*: Stringa opzionale utilizzata per indicare il criterio da utilizzare per elencare i nomi dei progetti di build. I valori validi includono:
 - `CREATED_TIME`: elenca i nomi dei progetti di compilazione in base alla data di creazione di ciascun progetto.
 - `LAST_MODIFIED_TIME`: elenca i nomi dei progetti di compilazione in base alle informazioni sull'ultima modifica apportata a ciascun progetto.
 - `NAME`: elenca i nomi dei progetti di compilazione in base al nome di ciascun progetto.
- *sort-order*: stringa opzionale utilizzata per indicare l'ordine in base al quale elencare i progetti di compilazione. *sort-by* I valori validi includono ASCENDING e DESCENDING.

- ***next-token***: stringa opzionale. Se durante un'esecuzione precedente c'erano più di 100 voci nell'elenco, vengono restituiti solo i primi 100 elementi, insieme a una stringa univoca denominata token successivo. Per ottenere il gruppo successivo di elementi nell'elenco, eseguire il comando nuovamente, aggiungendo il token successivo alla chiamata. Per ottenere tutti gli elementi dell'elenco, è necessario continuare a eseguire il comando con ogni token successivo restituito, finché non terminano i token successivi restituiti.

Ad esempio, se si esegue questo comando:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Se si esegue nuovamente il comando:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}
```

Visualizzazione di un elenco di nomi di progetti di compilazione (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, vedere [AWS SDKs e riferimento agli strumenti](#).

Si integra AWS CodeBuild

Una build rappresenta un insieme di azioni eseguite per creare artefatti di AWS CodeBuild output (ad esempio, un file JAR) basati su un insieme di artefatti di input (ad esempio, una raccolta di file di classe Java).

Le regole seguenti si applicano quando esegui più di una build:

- Se possibile, le build vengono eseguite simultaneamente. Il numero massimo di build in esecuzione simultanea può variare. Per ulteriori informazioni, consulta [Quote per AWS CodeBuild](#).
- Se il progetto di compilazione ha un limite di compilazione simultanea, le build restituiscono un errore se il numero di build in esecuzione raggiunge il limite di compilazione simultanea per il progetto. Per ulteriori informazioni, consulta [Abilita](#) il limite di compilazione simultanea.
- Se il progetto di compilazione non ha un limite di compilazione simultanea, le build vengono messe in coda se il numero di build in esecuzione raggiunge il limite di compilazione simultanea per la piattaforma e il tipo di calcolo. Il numero massimo di build in una coda è cinque volte il limite di build in esecuzione simultanea. Per ulteriori informazioni, consulta [Quote per AWS CodeBuild](#).

Se una build presente in una coda non viene avviata entro il numero di minuti specificato per il valore di timeout, viene rimossa dalla coda. Il valore predefinito del timeout è di 8 ore. Puoi sostituire il timeout della coda delle build con un valore compreso tra cinque minuti e otto ore quando esegui una build. Per ulteriori informazioni, consulta [Esegui AWS CodeBuild le build manualmente](#).

Non è possibile prevedere l'ordine in cui le build in coda vengono avviate.

Note

È possibile accedere alla cronologia di una build per un anno.

Puoi eseguire queste attività durante l'utilizzo di compilazioni:

Argomenti

- [Esegui AWS CodeBuild le build manualmente](#)
- [Esegui build su calcolo AWS Lambda](#)

- [Run si basa su flotte a capacità riservata](#)
- [Esegui le build in batch](#)
- [Esegui test paralleli in build in batch](#)
- [Creazioni di cache per migliorare le prestazioni](#)
- [Il debug si integra AWS CodeBuild](#)
- [Elimina build in AWS CodeBuild](#)
- [Retry si integra manualmente AWS CodeBuild](#)
- [Retry si integra automaticamente AWS CodeBuild](#)
- [Stop si integra AWS CodeBuild](#)
- [Interrompi gli accumuli in batch AWS CodeBuild](#)
- [Trigger AWS CodeBuild si crea automaticamente](#)
- [Visualizza i dettagli della costruzione in AWS CodeBuild](#)
- [Visualizza un elenco di build IDs in AWS CodeBuild](#)
- [Visualizza un elenco di build IDs per un progetto di compilazione in AWS CodeBuild](#)

Esegui AWS CodeBuild le build manualmente

Puoi usare la AWS CodeBuild console o AWS SDKs eseguire una build in CodeBuild. AWS CLI

Argomenti

- [Esegui le build localmente con l'agente AWS CodeBuild](#)
- [Esecuzione di una compilazione \(console\)](#)
- [Esecuzione di una compilazione \(AWS CLI\)](#)
- [Esegui una build in batch \(AWS CLI\)](#)
- [Avvio di esecuzioni di compilazione automaticamente \(AWS CLI\)](#)
- [Interruzione di esecuzioni di compilazioni automaticamente \(AWS CLI\)](#)
- [Esecuzione di una compilazione \(AWS SDKs\)](#)

Esegui le build localmente con l'agente AWS CodeBuild

È possibile utilizzare l' AWS CodeBuild agente per eseguire CodeBuild build su un computer locale. Sono disponibili agenti per piattaforme x86_64 e ARM.

È inoltre possibile abbonarsi per ricevere notifiche quando vengono pubblicate nuove versioni dell'agente.

Prerequisiti

Prima di iniziare, devi fare quanto segue:

- Installa Git sul computer locale.
- Installa e [configura Docker](#) sul tuo computer locale.

Configura l'immagine di build

È necessario configurare l'immagine di build solo la prima volta che si esegue l'agente o quando l'immagine è stata modificata.

Per configurare l'immagine di compilazione

1. [Se desideri utilizzare un'immagine Amazon Linux 2 curata, puoi recuperarla dal repository CodeBuild pubblico di Amazon ECR all'indirizzo https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard](https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standard) con il seguente comando:

```
$ docker pull public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0
```

In alternativa, se desideri utilizzare un'altra immagine Linux, procedi nel seguente modo:

- a. Clona il repository di CodeBuild immagini:

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Passa alla cartella delle immagini. Per questo esempio, usa l'`aws/codebuild/standard:5.0` immagine:

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Costruisci l'immagine. Questo processo può richiedere diversi minuti.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Scarica l' CodeBuild agente.

Per scaricare la versione x86_64 dell'agente, esegui il seguente comando:

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Per scaricare la versione ARM dell'agente, esegui il seguente comando:

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

3. L' CodeBuild agente è disponibile all'[indirizzo https://gallery.ecr.aws/codebuild/local-costruisce](https://gallery.ecr.aws/codebuild/local-costruisce).

La firma Secure Hash Algorithm (SHA) per la versione x86_64 dell'agente è:

```
sha256:ccb19bdd7af94e4dc761e4c58c267e9455c28ec68d938086b4dc1cf8fe6b0940
```

La firma SHA per la versione ARM dell'agente è:

```
sha256:7d7b5d35d2ac4e062ae7ba8c662ffed15229a52d09bd0d664a7816c439679192
```

È possibile utilizzare lo SHA per identificare la versione dell'agente. Per visualizzare la firma SHA dell'agente, esegui il comando seguente e cerca lo SHA in: RepoDigests

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

Esegui l'agente CodeBuild

Per gestire l' CodeBuild agente

1. Passa alla directory che contiene i sorgenti del progetto di compilazione.
2. Scarica lo script [codebuild_build.sh](#):

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/  
master/local_builds/codebuild_build.sh  
$ chmod +x codebuild_build.sh
```

3. Esegui lo `codebuild_build.sh` script e specifica l'immagine del contenitore e la directory di output.

Per eseguire una build x86_64, esegui il seguente comando:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Per eseguire una build ARM, esegui il seguente comando:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

Sostituisci *<container-image>* con il nome dell'immagine del contenitore, ad esempio `aws/codebuild/standard:5.0` o `public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0`.

Lo script avvia l'immagine di compilazione ed esegue la build sul progetto nella directory corrente. Per specificare la posizione del progetto di compilazione, aggiungete l'-s *<build project directory>* opzione al comando script.

Ricevi notifiche per le nuove versioni degli CodeBuild agenti

Puoi abbonarti alle notifiche di Amazon SNS in modo da ricevere una notifica quando vengono rilasciate nuove versioni dell' AWS CodeBuild agente.

Per sottoscrivere le notifiche degli CodeBuild agenti

1. [Apri la console Amazon SNS nella versione v3/home. https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. Nella barra di navigazione, se non è già selezionata, modifica la AWS regione in Stati Uniti orientali (Virginia settentrionale). Devi selezionare questa AWS regione perché le notifiche Amazon SNS a cui ti stai abbonando vengono create in questa regione.
3. Nel riquadro di navigazione, seleziona Sottoscrizioni.
4. Scegli Create Subscription (Crea sottoscrizione).
5. In Crea abbonamento, procedi come segue:

- a. Per ARN dell'argomento, utilizza il seguente Amazon Resource Name (ARN):

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. Per Protocollo, seleziona -mail) o SMS.

- c. Per Endpoint, scegli come ricevere le notifiche (e-mail o SMS). Inserisci un indirizzo e-mail o un numero di telefono, incluso il prefisso.
- d. Scegli Create Subscription (Crea sottoscrizione).
- e. Scegli Email per ricevere un'email che ti chiede di confermare l'iscrizione. Segui le istruzioni nell'e-mail per completare l'iscrizione.

Se non desideri più ricevere queste notifiche, segui la procedura seguente per annullare la sottoscrizione.

Per annullare l'iscrizione alle notifiche degli CodeBuild agenti

1. [Apri la console Amazon SNS nella versione v3/home. https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. Nel riquadro di navigazione, seleziona Sottoscrizioni.
3. Seleziona la sottoscrizione e da Actions (Operazioni), seleziona Delete subscriptions (Elimina sottoscrizioni). Quando viene richiesto di confermare, selezionare Delete (Elimina).

Esecuzione di una compilazione (console)

Per utilizzare AWS CodePipeline per eseguire una build con CodeBuild, salta questi passaggi e segui le istruzioni in [Usa CodeBuild con CodePipeline](#).

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Nell'elenco dei progetti di compilazione, scegli il progetto di compilazione.
4. Puoi eseguire la build con le impostazioni predefinite del progetto di compilazione o sostituire le impostazioni di compilazione solo per questa build.
 - a. Se desideri eseguire la build con le impostazioni predefinite del progetto di compilazione, scegli Avvia compilazione. La compilazione viene avviata immediatamente.
 - b. Se desideri sovrascrivere le impostazioni predefinite del progetto di compilazione, scegli Avvia compilazione con sostituzioni. Nella pagina Inizia la compilazione, puoi sovrascrivere quanto segue:
 - Configurazione della build
 - Origine

- Sostituzioni delle variabili di ambiente

Se devi selezionare sostituzioni di build più avanzate, scegli Sostituzioni di build avanzate. In questa pagina, puoi sovrascrivere quanto segue:

- Costruisci la configurazione
- Origine
- Ambiente
- Specifiche di costruzione
- Artefatti
- Log

Dopo aver effettuato le selezioni di override, scegli Avvia compilazione.

Per le informazioni dettagliate su questa compilazione, consultare [Visualizzazione dei dettagli delle compilazioni \(console\)](#).

Esecuzione di una compilazione (AWS CLI)

Note

Per utilizzare CodePipeline per eseguire una build con AWS CodeBuild, salta questi passaggi e segui le istruzioni in [Crea una pipeline che utilizza CodeBuild \(\)AWS CLI](#).

Per ulteriori informazioni sull'utilizzo di AWS CLI with CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

1. Eseguire il comando `start-build` in uno dei seguenti modi:

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

Utilizzare questo comando se si desidera eseguire una compilazione che utilizza la versione più recente dell'artefatto di input di compilazione e le impostazioni correnti del progetto di compilazione.

```
aws codebuild start-build --generate-cli-skeleton
```

Utilizzare questo comando se si desidera eseguire una compilazione con una versione precedente dell'artefatto di input di compilazione o se si desidera ignorare le impostazioni per gli artefatti di output di compilazione, le variabili di ambiente, la buildspec il periodo di timeout di compilazione predefinito.

2. Se eseguite il start-build comando con l'--project-nameopzione, sostituitelo *<project-name>* con il nome del progetto di compilazione, quindi passate al passaggio 6 di questa procedura. Per ottenere un elenco di progetti di compilazione, consulta [Visualizza i nomi dei progetti di compilazione](#).
3. Se esegui il start-build comando con l'--idempotency-tokenopzione, nella richiesta viene incluso un identificatore o token univoco con distinzione tra maiuscole e minuscole. start-build Il token è valido per 5 minuti dopo la richiesta . Se si ripete la start-build richiesta con lo stesso token, ma si modifica un parametro, viene CodeBuild restituito un errore di mancata corrispondenza del parametro.
4. Se si esegue il comando start-build con l'opzione --generate-cli-skeleton, i dati in formato JSON appaiono nell'output. Copia i dati in un file (ad esempio, *start-build.json*) in una posizione sul computer locale o sull'istanza in cui AWS CLI è installato. Modificare i dati copiati in modo che corrispondano a questo formato e salvare i risultati:

```
{
  "projectName": "projectName",
  "sourceVersion": "sourceVersion",
  "artifactsOverride": {
    "type": "type",
    "location": "location",
    "path": "path",
    "namespaceType": "namespaceType",
    "name": "artifactsOverride-name",
    "packaging": "packaging"
  },
  "buildspecOverride": "buildspecOverride",
  "cacheOverride": {
    "location": "cacheOverride-location",
    "type": "cacheOverride-type"
  },
  "certificateOverride": "certificateOverride",
  "computeTypeOverride": "computeTypeOverride",
```

```
"environmentTypeOverride": "environmentTypeOverride",
"environmentVariablesOverride": {
  "name": "environmentVariablesOverride-name",
  "value": "environmentVariablesValue",
  "type": "environmentVariablesOverride-type"
},
"gitCloneDepthOverride": "gitCloneDepthOverride",
"imageOverride": "imageOverride",
"idempotencyToken": "idempotencyToken",
"insecureSslOverride": "insecureSslOverride",
"privilegedModeOverride": "privilegedModeOverride",
"queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
"reportBuildStatusOverride": "reportBuildStatusOverride",
"timeoutInMinutesOverride": "timeoutInMinutesOverride",
"sourceAuthOverride": "sourceAuthOverride",
"sourceLocationOverride": "sourceLocationOverride",
"serviceRoleOverride": "serviceRoleOverride",
"sourceTypeOverride": "sourceTypeOverride"
}
```

Sostituire i seguenti segnaposto:

- **projectName**: stringa obbligatoria. Il nome del progetto di compilazione da utilizzare per questa compilazione.
- **sourceVersion**: stringa opzionale. Una versione del codice sorgente da creare, come segue:
 - Per Amazon S3, l'ID della versione che corrisponde alla versione del file ZIP di input che desideri creare. Se non **sourceVersion** è specificato, viene utilizzata la versione più recente.
 - Per CodeCommit, l'ID di commit che corrisponde alla versione del codice sorgente che desideri creare. Se non **sourceVersion** è specificato, viene utilizzato l'ID di commit HEAD del ramo predefinito. (Non è possibile specificare il nome del tag **sourceVersion**, ma è possibile specificare l'ID di commit del tag.)
 - Ad GitHub esempio, l'ID di commit, l'ID della pull request, il nome del ramo o il nome del tag che corrisponde alla versione del codice sorgente che desideri creare. Se si specifica l'ID di una richiesta pull, deve utilizzare il formato `pr/pull-request-ID` (ad esempio `pr/25`). Se un nome ramo è specificato, viene utilizzato l'ID commit HEAD del ramo. Se non **sourceVersion** è specificato, viene utilizzato l'ID di commit HEAD del ramo predefinito.

- Per Bitbucket, l'ID commit, il nome ramo o il nome del tag che corrisponde alla versione del codice sorgente che si desidera creare. Se un nome ramo è specificato, viene utilizzato l'ID commit HEAD del ramo. Se non *sourceVersion* è specificato, viene utilizzato l'ID di commit HEAD del ramo predefinito.
- I seguenti segnaposto sono per `artifactsOverride`.
 - *type*: facoltativo. Il tipo di artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *location*: facoltativo. La posizione dell'artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *path*: facoltativo. Il percorso di artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *namespaceType*: facoltativo. Il tipo di percorso di artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *name*: facoltativo. Il nome di artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *packaging*: facoltativo. Il tipo di pacchetto di artefatto di output di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.
 - *buildspecOverride*: facoltativo. Una dichiarazione di specifiche di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione. Se questo valore è impostato, può essere una definizione buildspec inlinea oppure il percorso a un file buildspec alternativo, relativo al valore della variabile di ambiente `CODEBUILD_SRC_DIR` incorporata o al percorso di un bucket S3. Il bucket S3 deve trovarsi nella stessa regione AWS del progetto build. Specificare il file buildspec utilizzando il relativo ARN (ad esempio, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Se tale valore non è fornito o è impostato su una stringa vuota, il codice di origine deve contenere un file `buildspec.yml` nella directory radice. Per ulteriori informazioni, consulta [Nome del file buildspec e posizione di storage](#).
- I seguenti segnaposto sono per `cacheOverride`.
 - *cacheOverride-location*: facoltativo. Il percorso di un oggetto ProjectCache per questa compilazione che sostituisce l'oggetto ProjectCache specificato nel progetto di compilazione. `cacheOverride` è facoltativo e richiede un oggetto ProjectCache. `location` è obbligatorio in un oggetto ProjectCache.
 - *cacheOverride-type*: facoltativo. Il tipo di un oggetto ProjectCache per questa compilazione che sostituisce l'oggetto ProjectCache specificato nel progetto di

compilazione. `cacheOverride` è facoltativo e richiede un oggetto `ProjectCache`. `type` è obbligatorio in un oggetto `ProjectCache`.

- **`certificateOverride`**: facoltativo. Il nome di un certificato per questa compilazione che sostituisce quella definita nel progetto di compilazione.
- **`environmentTypeOverride`**: facoltativo. Un tipo di container per questa compilazione che sostituisce quella definita nel progetto di compilazione. L'attuale stringa valida è `LINUX_CONTAINER`.
- I seguenti segnaposto sono per `environmentVariablesOverride`.
 - **`environmentVariablesOverride-name`**: facoltativo. Il nome di una variabile di ambiente nel progetto di compilazione il quale valore si desidera ignorare per questa compilazione.
 - **`environmentVariablesOverride-type`**: facoltativo. Il tipo di variabile di ambiente nel progetto di compilazione il quale valore si desidera ignorare per questa compilazione.
 - **`environmentVariablesValue`**: facoltativo. Il valore della variabile di ambiente definita nel progetto di compilazione il quale valore si desidera ignorare per questa compilazione.
- **`gitCloneDepthOverride`**: facoltativo. Il valore di Git clone depth nel progetto di compilazione il quale valore si desidera ignorare per questa compilazione. Se il tipo di origine è Amazon S3, questo valore non è supportato.
- **`imageOverride`**: facoltativo. Il nome di un'immagine per questa compilazione che sostituisce quella definita nel progetto di compilazione.
- **`idempotencyToken`**: facoltativo. Una stringa che serve come token per specificare che la richiesta di compilazione è idempotente. È possibile scegliere qualsiasi stringa con 64 caratteri o meno. Il token è valido per 5 minuti dopo la richiesta di `start-build`. Se ripeti la richiesta di `start-build` con lo stesso token, ma modifichi un parametro, CodeBuild restituisce un errore di mancata corrispondenza del parametro.
- **`insecureSslOverride`**: valore booleano facoltativo che specifica se sovrascrivere l'impostazione TLS non sicura specificata nel progetto di compilazione. L'impostazione TLS non sicura determina se ignorare gli avvisi TLS durante la connessione al codice sorgente del progetto. Questo override si applica solo se l'origine della build è Enterprise Server. GitHub
- **`privilegedModeOverride`**: valore booleano facoltativo. Se impostato su `true`, la compilazione sostituisce la modalità privilegiata nel progetto di compilazione.
- **`queuedTimeoutInMinutesOverride`**: numero intero opzionale che specifica il numero di minuti in cui una build può essere messa in coda prima del timeout. Il valore minimo è cinque minuti e il valore massimo è 480 minuti (8 ore).

- ***reportBuildStatusOverride***: Valore booleano facoltativo che specifica se inviare al provider di origine lo stato di inizio e completamento di una build. Se lo imposti con un provider di origine diverso GitHub da GitHub Enterprise Server o Bitbucket, viene generato un `invalidInputException`
- ***sourceAuthOverride***: stringa opzionale. Un tipo di autorizzazione per questa compilazione che sostituisce quella definita nel progetto di compilazione. Questo override si applica solo se l'origine del progetto di compilazione è Bitbucket o. GitHub
- ***sourceLocationOverride***: stringa opzionale. Il percorso che, per questa compilazione, sostituisce il percorso di origine definito nel progetto di compilazione.
- ***serviceRoleOverride***: stringa opzionale. Il nome di un ruolo del servizio per questa compilazione che sostituisce quella definita nel progetto di compilazione.
- ***sourceTypeOverride***: stringa opzionale. Un tipo di input di origine per questa compilazione che sostituisce l'input di origine definito nel progetto di compilazione. Le stringhe valide sono `NO_SOURCE`, `CODECOMMIT`, `CODEPIPELINE`, `GITHUB`, `S3`, `BITBUCKET` e `GITHUB_ENTERPRISE`.
- ***timeoutInMinutesOverride***: numero opzionale. Il numero di minuti di timeout di compilazione che, per questa compilazione, sostituisce quella definita nel progetto di compilazione.

Ti consigliamo di memorizzare una variabile di ambiente con un valore sensibile, come un ID di chiave di AWS accesso, una chiave di accesso AWS segreta o una password come parametro in Amazon EC2 Systems Manager Parameter Store. CodeBuild può utilizzare un parametro memorizzato in Amazon EC2 Systems Manager Parameter Store solo se il nome di quel parametro inizia con `/CodeBuild/` (ad esempio, `/CodeBuild/dockerLoginPassword`). Puoi utilizzare la CodeBuild console per creare un parametro in Amazon EC2 Systems Manager. Selezionare **Create a parameter** (Crea un parametro) e seguire le istruzioni. (In quella finestra di dialogo, per la chiave KMS, puoi facoltativamente specificare l'ARN di una AWS KMS chiave nel tuo account. Amazon EC2 Systems Manager utilizza questa chiave per crittografare il valore del parametro durante l'archiviazione e decrittografarlo durante il recupero.) Se usi la CodeBuild console per creare un parametro, la console avvia il parametro con il `/CodeBuild/` momento in cui viene memorizzato. Tuttavia, se utilizzi la console Amazon EC2 Systems Manager Parameter Store per creare un parametro, devi iniziare il nome del parametro con `/CodeBuild/` e devi impostare **Type** su **Secure String**. Per ulteriori informazioni, consulta [AWS Systems Manager Parameter Store](#) and [Walkthrough: Create and test a String parameter \(console\)](#) nella Amazon EC2 Systems Manager User Guide.

Se il progetto di compilazione si riferisce a parametri memorizzati in Amazon EC2 Systems Manager Parameter Store, il ruolo di servizio del progetto di compilazione deve consentire l'azione `ssm:GetParameters`. Se in precedenza hai scelto Crea un nuovo ruolo di servizio nel tuo account, CodeBuild include automaticamente questa azione nel ruolo di servizio predefinito per il tuo progetto di build. Tuttavia, se era stata scelta l'opzione Choose an existing service role from your account (Scegli un ruolo del servizio esistente dal tuo account), è necessario includere questa operazione al ruolo del servizio separatamente.

Le variabili di ambiente impostate sostituiscono quelle esistenti. Ad esempio, se l'immagine Docker contiene già una variabile di ambiente denominata `MY_VAR` con un valore di `my_value`, e si imposta una variabile di ambiente denominata `MY_VAR` con un valore di `other_value`, il valore `my_value` viene sostituito da `other_value`. Allo stesso modo, se l'immagine Docker contiene già una variabile di ambiente denominata `PATH` con un valore di `/usr/local/sbin:/usr/local/bin`, e si imposta una variabile di ambiente denominata `PATH` con un valore di `$PATH:/usr/share/ant/bin`, il valore di `/usr/local/sbin:/usr/local/bin` viene sostituito dal valore letterale `$PATH:/usr/share/ant/bin`.

Non impostare variabili di ambiente con nomi che iniziano con `CODEBUILD_`. Questo prefisso è riservato per l'uso interno.

Se una variabile di ambiente con lo stesso nome viene definita in più luoghi, il valore della variabile di ambiente viene determinato come segue:

- Il valore della chiamata all'operazione di inizio della compilazione ha la massima priorità.
- Il valore della definizione del progetto di compilazione è il seguente in ordine di priorità.
- Il valore della dichiarazione `buildspec` ha la minima priorità.

Per informazioni sui valori validi per questi segnaposto, consultare [Creazione di un progetto di compilazione \(AWS CLI\)](#). Per un elenco delle impostazioni più recenti per un progetto di compilazione, consultare [Visualizza i dettagli del progetto di costruzione](#).

5. Passare alla directory contenente il file appena salvato ed eseguire nuovamente il comando `start-build`.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. In caso di esito positivo, nell'output vengono visualizzati dati simili a quelli descritti nella procedura [Per eseguire la build](#).

Per utilizzare le informazioni dettagliate su questa compilazione, annotare il valore `id` nell'output e consultare [Visualizzazione dei dettagli delle compilazioni \(AWS CLI\)](#).

Esegui una build in batch (AWS CLI)

1. Eseguire il comando `start-build-batch` in uno dei seguenti modi:

```
aws codebuild start-build-batch --project-name <project-name>
```

Utilizzare questo comando se si desidera eseguire una compilazione che utilizza la versione più recente dell'artefatto di input di compilazione e le impostazioni correnti del progetto di compilazione.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Utilizzare questo comando se si desidera eseguire una compilazione con una versione precedente dell'artefatto di input di compilazione o se si desidera ignorare le impostazioni per gli artefatti di output di compilazione, le variabili di ambiente, la buildspec il periodo di timeout di compilazione predefinito.

2. Se eseguite il `start-build-batch` comando con l'`--project-name` opzione, sostituitelo *<project-name>* con il nome del progetto di compilazione, quindi passate al passaggio 6 di questa procedura. Per ottenere un elenco di progetti di compilazione, consulta [Visualizza i nomi dei progetti di compilazione](#).
3. Se esegui il `start-build-batch` comando con l'`--idempotency-token` opzione, nella richiesta viene incluso un identificatore o token univoco con distinzione tra maiuscole e minuscole. `start-build-batch` Il token è valido per 5 minuti dopo la richiesta. Se si ripete la `start-build-batch` richiesta con lo stesso token, ma si modifica un parametro, viene CodeBuild restituito un errore di mancata corrispondenza del parametro.
4. Se si esegue il `start-build-batch` comando con l'`--generate-cli-skeleton` opzione, i dati in formato JSON vengono emessi nel file. *<json-file>* Questo file è simile allo skelton prodotto dal `start-build` comando, con l'aggiunta del seguente oggetto. Per ulteriori informazioni sugli oggetti comuni, vedere. [Esecuzione di una compilazione \(AWS CLI\)](#)

Modifica questo file per aggiungere eventuali sostituzioni di build e salva i risultati.

```
"buildBatchConfigOverride": {  
  "combineArtifacts": combineArtifacts,
```

```
"restrictions": {
  "computeTypesAllowed": [
    allowedComputeTypes
  ],
  "maximumBuildsAllowed": maximumBuildsAllowed
},
"serviceRole": "batchServiceRole",
"timeoutInMins": batchTimeout
}
```

L'`buildBatchConfigOverride` oggetto è una [ProjectBuildBatchConfig](#) struttura che contiene le modifiche di configurazione della build in batch per questa build.

combineArtifacts

Un valore booleano che specifica se gli elementi di compilazione per la compilazione in batch devono essere combinati in un'unica posizione degli artefatti.

allowedComputeTypes

Array di stringhe che specificano i tipi di calcolo consentiti per la compilazione batch. Vedi Tipi di calcolo dell'ambiente di [compilazione per questi valori](#).

maximumBuildsAllowed

Specifica il numero massimo di compilazioni consentite.

batchServiceRole

Specifica l'ARN del ruolo di servizio per il progetto di compilazione batch.

batchTimeout

Specifica la quantità massima di tempo, espressa in minuti, in cui deve essere completata la compilazione batch.

5. Passare alla directory contenente il file appena salvato ed eseguire nuovamente il comando `start-build-batch`.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. In caso di successo, la rappresentazione JSON di un [BuildBatch](#) oggetto viene visualizzata nell'output della console. Vedi la [sintassi di StartBuildBatch risposta](#) per un esempio di questi dati.

Avvio di esecuzioni di compilazione automaticamente (AWS CLI)

Se il codice sorgente è archiviato in un repository GitHub o in un repository GitHub Enterprise Server, è possibile utilizzare i GitHub webhook per AWS CodeBuild ricostruire il codice sorgente ogni volta che viene inserita una modifica al codice nell'archivio.

Eseguire il comando `create-webhook` nel modo seguente:

```
aws codebuild create-webhook --project-name <project-name>
```

<project-name> è il nome del progetto di compilazione che contiene il codice sorgente da ricostruire.

Infatti GitHub, nell'output vengono visualizzate informazioni simili alle seguenti:

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

<url> è l'URL del GitHub webhook.

Per GitHub Enterprise Server, nell'output vengono visualizzate informazioni simili alle seguenti:

```
{
  "webhook": {
    "secret": "YRV4JYAGFsekJiirp5ytx86oZpyhUdySNSDTLNuXoXX1c7aZ6XYDf37-ZFyY02rs4JSE70mLW3w-gh-ryoVB80SSSC1aAtBtuPkHwYuncCCmdogCVCfniQ7ukYX2_xM--n1Dma5EngIg_Bi_N465yi33zyTUNPoQ1xCpL0-BwghcVa9lAurwR77-uY7i-_XCJFahwMx1f4ubOgBB5mMT2A16apqjQJoKSb61XVKyZy1Giuy4nliAXfv9Wnn76CaCsndb3fVIE78fpygfo41xYxSQ6vpo6LRTKtPzbyeTHbVXGda1PJvnbBlnKmJDo0RTgI1m2oYr17dWziQ1rrvoCoNgy1SOO_7LKfA-nNXFc_f1SiFy0AqeMB43-d00cdkzybHncE81QTRwEUCFfmX-AJCwmlXV0kg0G67T92Sjbpz0fRlkh5pwIF193_b8_jOHDinK6iOiPpf2dIDAIZgGMagqZeWb-axDeTAbopoU8J6gFI1yKo5aq9q151zC1PERUsMgJfTjr_a-Z-L_kylr-4hSSxasS3NuJ43_X0BRWqT51xqvH-A69bV07KbVT_Kc6wxkSHyYCEMoa_Pfa7ZQgyfY6B00ogMNj31yFbjthORN1cDo6-3J-McDLoYrRtSEOV9QnxvsG5zu1N5-z20rkJtg_MOfNwocfUutFXb7vrGTduH1R1dzXLRushuXOVVuDUwm9vhwMnr-hUkeGo_1kDKyk4E2QFvZXPjYw0vFfV-dwxFRR_mifzxW1wyfmt2iFtLkp_YZj_4WeFAckGefr-ilNaYvsZpzXj78Ae1adVoLf48AmDdN2pWslWJjatU9zt942gLiSFFmKakcvJuy5yxxHaxxbhUyC8NHYIESUWPfcfnqrMsr8op3P4AUCHIpiZCYUiwI_cac-pIUB00Xaur_lu_fyFghgOJc7cfTnA36rv5X5DnFDM8P3HNBELjaF9QZ6AiejegPEWTHIKJON3AUDwpkz_hwTXyUoAU8MdZfPTXbBoT6N5Z5THBHsYxR",
    "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?t=eyJlbmNyeXB0ZWREYXRhIjoiaUmfqMmJERGRQbGhwLzNTNld3R0VGRjZzOTNwLz1ZVG1NZ1pIR1E0RUsxdzhGeWhnVFFqWTR0WEFwT2dJRnNmRHc3S3RNc0xYMEncXFTakg1cE1nSy9zPSIsIm12UGFyYW1ldGVyU3B1YyI6IndSQ1Qrc2VPOjBCZzhPeVYiLCJtYXR1cm1hbFNldFN1cm1hbCI6MX0%3D&v=1"
  }
}
```

1. Copiare l'URL della chiave segreta e di payload dall'output. Sono necessari per aggiungere un webhook in GitHub Enterprise Server.
2. In GitHub Enterprise Server, scegliete il repository in cui è archiviato il CodeBuild progetto. Selezionare Settings (Impostazioni), quindi Hooks & services (Hook e servizi) e selezionare Add webhook (Aggiungi webhook).

3. Inserire l'URL di payload e della chiave segreta, accettare le impostazioni predefinite per gli altri campi e selezionare Add webhook (Aggiungi webhook).

Interruzione di esecuzioni di compilazioni automaticamente (AWS CLI)

Se il codice sorgente è archiviato in un repository GitHub o in un repository GitHub Enterprise Server, è possibile configurare dei GitHub webhook in modo da AWS CodeBuild ricostruire il codice sorgente ogni volta che viene inserita una modifica al codice nell'archivio. Per ulteriori informazioni, consulta [Avvio di esecuzioni di compilazione automaticamente \(AWS CLI\)](#).

Se hai abilitato questo comportamento, è possibile disattivarlo eseguendo il comando delete-webhook come segue:

```
aws codebuild delete-webhook --project-name <project-name>
```

- dove *<project-name>* è il nome del progetto di compilazione che contiene il codice sorgente da ricostruire.

Se questo comando viene completato, nessuna informazione e nessun errore appaiono nell'output.

Note

Questo elimina il webhook solo dal tuo CodeBuild progetto. È inoltre necessario eliminare il webhook dal proprio repository GitHub o da quello di GitHub Enterprise Server.

Esecuzione di una compilazione (AWS SDKs)

Per utilizzare CodePipeline per eseguire una build con AWS CodeBuild, salta questi passaggi e segui invece le istruzioni riportate. [Usa AWS CodeBuild with AWS CodePipeline per testare il codice ed eseguire build](#)

Per informazioni sull'utilizzo CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Esegui build su calcolo AWS Lambda

AWS Lambda compute offre velocità di avvio ottimizzate per le tue build. AWS Lambda supporta build più veloci grazie a una minore latenza di avvio. AWS Lambda inoltre si ridimensiona automaticamente, quindi le build non attendono in coda per essere eseguite. Tuttavia, ci sono alcuni casi d'uso che AWS Lambda non sono supportati e, se hanno un impatto su di te, usa il calcolo EC2. Per ulteriori informazioni, consulta [Limitazioni di AWS Lambda calcolo](#).

Argomenti

- [Quali strumenti e runtime saranno inclusi nelle immagini docker dell'ambiente di runtime curato su cui vengono eseguiti? AWS Lambda](#)
- [Cosa succede se l'immagine curata non include gli strumenti di cui ho bisogno?](#)
- [In quali aree geografiche è AWS Lambda supportata l'elaborazione? CodeBuild](#)
- [Limitazioni di AWS Lambda calcolo](#)
- [Implementa una funzione Lambda utilizzando AWS SAM con Lambda Java CodeBuild](#)
- [Crea un'app React a pagina singola con CodeBuild Lambda Node.js](#)
- [Aggiornare la configurazione di una funzione Lambda con CodeBuild Lambda Python](#)

Quali strumenti e runtime saranno inclusi nelle immagini docker dell'ambiente di runtime curato su cui vengono eseguiti? AWS Lambda

AWS Lambda supporta i seguenti strumenti: AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby e.NET.

Cosa succede se l'immagine curata non include gli strumenti di cui ho bisogno?

Se l'immagine curata non include gli strumenti necessari, puoi fornire un'immagine Docker di ambiente personalizzata che includa gli strumenti necessari.

Note

Lambda non supporta funzioni che utilizzano immagini container multi-architettura. Per ulteriori informazioni, consulta [Creare una funzione Lambda utilizzando un'immagine del contenitore nella Guida](#) per gli AWS Lambda sviluppatori.

Tieni presente che sono necessarie le seguenti autorizzazioni Amazon ECR per utilizzare immagini personalizzate per il calcolo Lambda:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/image-repo"
    }
  ]
}
```

Tieni inoltre presente che `curl` o `wget` deve essere installato per poter utilizzare immagini personalizzate.

In quali aree geografiche è AWS Lambda supportata l'elaborazione? CodeBuild

Nel CodeBuild, l'AWS Lambda elaborazione è supportata nei seguenti paesi Regioni AWS: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Mumbai), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte), Europa (Irlanda) e Sud America (San Paolo). [Per ulteriori informazioni su Regioni AWS dove CodeBuild è disponibile, consulta Servizi per regione.AWS](#)

Limitazioni di AWS Lambda calcolo

Esistono alcuni casi d'uso che AWS Lambda non sono supportati e, se hanno un impatto su di te, usa il calcolo EC2:

- AWS Lambda non supporta strumenti che richiedono i permessi di root. Per strumenti come `yum` o `rpm`, usa il tipo di calcolo EC2 o altri strumenti che non richiedono i permessi di root.
- AWS Lambda non supporta le build o le esecuzioni di Docker.
- AWS Lambda non supporta la scrittura su file esterni. `/tmp` I gestori di pacchetti inclusi sono configurati per utilizzare la `/tmp` directory di default per scaricare e fare riferimento ai pacchetti.
- AWS Lambda non supporta il tipo di ambiente `LINUX_GPU_CONTAINER` e non è supportato su Windows Server Core 2019.
- AWS Lambda non supporta la memorizzazione nella cache, i timeout di compilazione personalizzati, il timeout della coda, i badge di creazione, la modalità privilegiata, gli ambienti di runtime personalizzati o i runtime superiori a 15 minuti.
- AWS Lambda non supporta la connettività VPC, un intervallo fisso di indirizzi IP di CodeBuild origine, EFS, l'installazione di certificati o l'accesso SSH con Session Manager.

Implementa una funzione Lambda utilizzando AWS SAM con Lambda Java CodeBuild

Il AWS Serverless Application Model (AWS SAM) è un framework open source per la creazione di applicazioni serverless. Per ulteriori informazioni, consulta il [AWS Serverless Application Model repository](#) su GitHub. Il seguente esempio di Java utilizza Gradle per creare e testare una AWS Lambda funzione. Dopodiché, la AWS SAM CLI viene utilizzata per distribuire il CloudFormation modello e il pacchetto di distribuzione. Utilizzando CodeBuild Lambda, le fasi di compilazione, test e implementazione vengono tutte gestite automaticamente, consentendo di aggiornare rapidamente l'infrastruttura senza l'intervento manuale in un'unica build.

Configura il tuo repository AWS SAM

Crea un AWS SAM Hello World progetto utilizzando la AWS SAM CLI.

Per creare il tuo progetto AWS SAM

1. Segui le istruzioni contenute nella Guida per gli AWS Serverless Application Model sviluppatori per [l'installazione della AWS SAM CLI](#) sul tuo computer locale.

2. Esegui `sam init` e seleziona la seguente configurazione del progetto.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions?
[y/N]: N
Project name [sam-app]: <insert project name>
```

3. Carica la cartella AWS SAM del progetto in un archivio di sorgenti supportato. Per un elenco dei tipi di sorgenti supportati, consulta [ProjectSource](#).

Creare un progetto CodeBuild Lambda Java

Crea un progetto AWS CodeBuild Lambda Java e configura le autorizzazioni IAM necessarie per la build.

Per creare il tuo progetto CodeBuild Lambda Java

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
4. In Source, seleziona il repository di origine in cui si trova il AWS SAM progetto.
5. In Environment (Ambiente):
 - Per Compute, seleziona Lambda.
 - Per Runtime (s), seleziona Java.
 - Per Image, selezionate `aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21`.

- Per Ruolo di servizio, lascia selezionata l'opzione Nuovo ruolo di servizio. Prendi nota del nome del ruolo. Ciò sarà necessario quando aggiornerai le autorizzazioni IAM del progetto più avanti in questo esempio.
6. Scegliere Create build project (Crea progetto di compilazione).
 7. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 8. Nel riquadro di navigazione, scegli Ruoli e seleziona il ruolo di servizio associato al tuo progetto. Puoi trovare il tuo ruolo nel progetto CodeBuild selezionando il progetto di compilazione, scegliendo Modifica, Ambiente e quindi Ruolo di servizio.
 9. Scegli la scheda Relazioni di attendibilità e quindi Modifica policy di attendibilità.
 10. Aggiungi la seguente policy in linea al tuo ruolo IAM. Questo verrà utilizzato per implementare la tua AWS SAM infrastruttura in un secondo momento. Per ulteriori informazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente di IAM .

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "iam:*",
        "apigateway:*",
        "s3:*"
      ],
      "Resource": "arn:aws:iam::*:role/Service*"
    }
  ]
}
```

Configura il progetto buildspec

Per creare, testare e distribuire la tua funzione Lambda CodeBuild , legge ed esegue i comandi di compilazione da un buildspec.

Per configurare il tuo progetto buildspec

1. Nella CodeBuild console, seleziona il tuo progetto di build, quindi scegli Modifica e Buildspec.
2. In Buildspec, scegli Inserisci comandi di compilazione e poi Passa all'editor.
3. Elimina i comandi di compilazione precompilati e incollali nella seguente buildspec.

```
version: 0.2
env:
  variables:
    GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
      - sam build --template-file template.yaml
      - echo "Running deploy..."
      - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
      - yes | sam deploy
```

4. Selezionare Update buildspec (Aggiorna buildspec).

Implementa la tua infrastruttura AWS SAM Lambda

Usa CodeBuild Lambda per implementare automaticamente la tua infrastruttura Lambda

Per implementare la tua infrastruttura Lambda

1. Selezionare Start build (Avvia compilazione). Questo creerà, testerà e distribuirà automaticamente AWS SAM l'applicazione per utilizzarla. AWS Lambda CloudFormation
2. Una volta terminata la compilazione, accedi alla AWS Lambda console e cerca la tua nuova funzione Lambda sotto il nome del AWS SAM progetto.
3. Testa la tua funzione Lambda selezionando API Gateway nella panoramica delle funzioni, quindi facendo clic sull'URL dell'endpoint API. Dovresti vedere una pagina aperta con il messaggio.
"message": "hello world"

Pulisci la tua infrastruttura

Per evitare ulteriori addebiti per le risorse utilizzate durante questo tutorial, elimina le risorse create dal AWS SAM modello e CodeBuild.

Per ripulire la tua infrastruttura

1. Vai alla CloudFormation console e seleziona `aws-sam-cli-managed-default`.
2. In Risorse, svuota il bucket `SamCliSourceBucket` di distribuzione.
3. Elimina lo `aws-sam-cli-managed-default` stack.
4. Elimina lo CloudFormation stack associato al tuo AWS SAM progetto. Questo stack dovrebbe avere lo stesso nome del tuo AWS SAM progetto.
5. Accedi alla CloudWatch console ed elimina i gruppi di CloudWatch log associati al tuo CodeBuild progetto.
6. Vai alla CodeBuild console ed elimina il CodeBuild progetto scegliendo Elimina progetto di compilazione.

Crea un'app React a pagina singola con CodeBuild Lambda Node.js

[Create React App](#) è un modo per creare applicazioni React a pagina singola. Il seguente esempio di Node.js utilizza Node.js per creare gli artefatti di origine dall'app Create React e restituisce gli artefatti di compilazione.

Configura il repository dei sorgenti e il bucket degli artefatti

Crea un archivio di sorgenti per il tuo progetto usando yarn e l'app Create React.

Per configurare il repository dei sorgenti e il bucket degli artefatti

1. Sul tuo computer locale, esegui `yarn create react-app <app-name>` per creare una semplice app React.
2. Carica la cartella del progetto dell'app React in un repository di sorgenti supportato. Per un elenco dei tipi di sorgenti supportati, vedi [ProjectSource](#).

Creare un progetto CodeBuild Lambda Node.js

Crea un progetto AWS CodeBuild Lambda Node.js.

Per creare il tuo progetto CodeBuild Lambda Node.js

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.
3. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
4. In Source, seleziona il repository di origine in cui si trova il AWS SAM progetto.
5. In Environment (Ambiente):
 - Per Compute, seleziona Lambda.
 - Per Runtime (s), seleziona Node.js.
 - Per Image, selezionate aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.
6. In Artifacts (Artefatti):
 - Per Tipo, seleziona Amazon S3.
 - Per Bucket name, seleziona il bucket di artefatti del progetto che hai creato in precedenza.
 - Per il packaging degli artifatti, seleziona Zip.
7. Scegliere Create build project (Crea progetto di compilazione).

Configura il progetto buildspec

Per creare la tua app React, CodeBuild legge ed esegue i comandi di compilazione da un file buildspec.

Per configurare il tuo progetto buildspec

1. Nella CodeBuild console, seleziona il tuo progetto di build, quindi scegli Modifica e Buildspec.
2. In Buildspec, scegli Inserisci comandi di compilazione e poi Passa all'editor.
3. Elimina i comandi di compilazione precompilati e incollali nella seguente buildspec.

```
version: 0.2
phases:
```

```
build:
  commands:
    - yarn
    - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
    - yarn run build
    - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
  artifacts:
    name: "build-output"
    files:
      - "**/*"
  reports:
    test-report:
      files:
        - 'junit.xml'
      file-format: 'JUNITXML'
    coverage-report:
      files:
        - 'coverage/clover.xml'
      file-format: 'CLOVERXML'
```

4. Selezionare Update buildspec (Aggiorna buildspec).

Crea ed esegui la tua app React

Crea l'app React su CodeBuild Lambda, scarica gli artefatti di compilazione ed esegui l'app React localmente.

Per creare ed eseguire la tua app React

1. Selezionare Start build (Avvia compilazione).
2. Una volta terminata la compilazione, accedi al tuo bucket di artefatti del progetto Amazon S3 e scarica l'artefatto dell'app React.
3. Decomprimi l'artefatto della build di React e inseriscilo nella cartella del progetto. `run npm install -g serve && serve -s build`
4. Il serve comando servirà il sito statico su una porta locale e stamperà l'output sul tuo terminale. Puoi visitare l'URL localhost sotto Local: nell'output del terminale per visualizzare la tua app React.

Per saperne di più su come gestire la distribuzione di un server basato su React, vedi [Create React App Deployment](#).

Pulisci la tua infrastruttura

Per evitare ulteriori addebiti per le risorse utilizzate durante questo tutorial, elimina le risorse create per il CodeBuild progetto.

Per ripulire la tua infrastruttura

1. Eliminare gli artefatti del progetto (bucket Amazon S3)
2. Accedi alla CloudWatch console ed elimina i gruppi di CloudWatch log associati al tuo progetto. CodeBuild
3. Vai alla CodeBuild console ed elimina il CodeBuild progetto scegliendo Elimina progetto di compilazione.

Aggiornare la configurazione di una funzione Lambda con CodeBuild Lambda Python

Il seguente esempio di Python utilizza [Boto3](#) e Lambda Python per aggiornare la configurazione di una funzione CodeBuild Lambda. Questo esempio può essere esteso per gestire altre risorse a livello di codice. AWS Per ulteriori informazioni, consultate la documentazione di [Boto3](#).

Prerequisiti

Crea o trova una funzione Lambda nel tuo account.

Questo esempio presuppone che tu abbia già creato una funzione Lambda nel tuo account e che la CodeBuild utilizzerai per aggiornare le variabili di ambiente della funzione Lambda. Per ulteriori informazioni sulla configurazione di una funzione Lambda CodeBuild, guarda l'[Implementa una funzione Lambda utilizzando AWS SAM con Lambda Java CodeBuild](#) esempio o visita. [AWS Lambda](#)

Configura il tuo repository dei sorgenti

Crea un repository di sorgenti per archiviare lo script python Boto3.

Per configurare il repository dei sorgenti

1. Copia il seguente script python in un nuovo file chiamato. `update_lambda_environment_variables.py`


```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
    print("Updating lambda function " + lambda_function_name + " environment
variable "
        + lambda_env_variable + " to " + lambda_env_variable_value)
    lambda_client.update_function_configuration(
        FunctionName=lambda_function_name,
        Environment={
            'Variables': {
                lambda_env_variable: lambda_env_variable_value
            }
        },
    )

if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Carica il file python in un repository di sorgenti supportato. Per un elenco dei tipi di sorgenti supportati, consulta. [ProjectSource](#)

Crea un progetto CodeBuild Lambda Python

Crea un progetto CodeBuild Lambda Python.

Per creare il tuo progetto CodeBuild Lambda Java

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Se viene visualizzata una pagina di CodeBuild informazioni, scegli Crea progetto di compilazione. Altrimenti, nel riquadro di navigazione, espandi Crea, scegli Crea progetti, quindi scegli Crea progetto di compilazione.

3. In Project name (Nome progetto) immettere un nome per questo progetto di compilazione. I nomi dei progetti di compilazione devono essere univoci per ogni AWS account. Puoi anche includere una descrizione opzionale del progetto di compilazione per aiutare gli altri utenti a capire a cosa serve questo progetto.
4. In Source, seleziona il repository di origine in cui si trova il AWS SAM progetto.
5. In Environment (Ambiente):
 - Per Compute, seleziona Lambda.
 - Per Runtime (s), selezionate Python.
 - Per Image, seleziona aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12.
 - Per Ruolo di servizio, lascia selezionata l'opzione Nuovo ruolo di servizio. Prendi nota del nome del ruolo. Ciò sarà necessario quando aggiornerai le autorizzazioni IAM del progetto più avanti in questo esempio.
6. Scegliere Create build project (Crea progetto di compilazione).
7. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
8. Nel riquadro di navigazione, scegli Ruoli e seleziona il ruolo di servizio associato al tuo progetto. Puoi trovare il tuo ruolo nel progetto CodeBuild selezionando il progetto di compilazione, scegliendo Modifica, Ambiente e quindi Ruolo di servizio.
9. Scegli la scheda Relazioni di attendibilità e quindi Modifica policy di attendibilità.
10. Aggiungi la seguente policy in linea al tuo ruolo IAM. Questo verrà utilizzato per implementare la tua AWS SAM infrastruttura in un secondo momento. Per ulteriori informazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente di IAM .

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateLambdaPermissions",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
    ]  
  }  
]  
}
```

Configura il progetto buildspec

Per aggiornare la funzione Lambda, lo script legge le variabili di ambiente da buildspec per trovare il nome della funzione Lambda, il nome della variabile di ambiente e il valore della variabile di ambiente.

Per configurare il tuo progetto buildspec

1. Nella CodeBuild console, seleziona il tuo progetto di build, quindi scegli Modifica e Buildspec.
2. In Buildspec, scegli Inserisci comandi di compilazione e poi Passa all'editor.
3. Elimina i comandi di compilazione precompilati e incollali nella seguente buildspec.

```
version: 0.2  
env:  
  variables:  
    LAMBDA_FUNC_NAME: "<lambda-function-name>"  
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"  
    LAMBDA_ENV_VARIABLE_VALUE: "true"  
phases:  
  install:  
    commands:  
      - pip3 install boto3  
  build:  
    commands:  
      - python3 update_lambda_environment_variables.py
```

4. Selezionare Update buildspec (Aggiorna buildspec).

Aggiorna la tua configurazione Lambda

Usa CodeBuild Lambda Python per aggiornare automaticamente la configurazione della tua funzione Lambda.

Per aggiornare la configurazione della funzione Lambda

1. Selezionare Start build (Avvia compilazione).
2. Una volta terminata la compilazione, accedi alla tua funzione Lambda.
3. Seleziona Configurazione e poi Variabili di ambiente. Dovresti vedere una nuova variabile di ambiente con chiave `FEATURE_ENABLED` e valore `true`.

Pulisci la tua infrastruttura

Per evitare ulteriori addebiti per le risorse utilizzate durante questo tutorial, elimina le risorse create per il CodeBuild progetto.

Per ripulire la tua infrastruttura

1. Accedi alla CloudWatch console ed elimina i gruppi di CloudWatch log associati al tuo CodeBuild progetto.
2. Vai alla CodeBuild console ed elimina il tuo CodeBuild progetto scegliendo Elimina progetto di compilazione.
3. Se hai creato una funzione Lambda ai fini di questo esempio, scegli la funzione Azioni ed Elimina per ripulire la funzione Lambda.

Estensioni

Se vuoi estendere questo esempio per gestire altre AWS risorse usando AWS CodeBuild Lambda Python:

- Aggiorna lo script Python per modificare le nuove risorse usando Boto3.
- Aggiorna il ruolo IAM associato al tuo CodeBuild progetto per disporre delle autorizzazioni per le nuove risorse.
- Aggiungi eventuali nuove variabili di ambiente associate alle nuove risorse al tuo buildspec.

Run si basa su flotte a capacità riservata

CodeBuild offre le seguenti flotte di elaborazione:

- flotte su richiesta

- Flotte a capacità riservata

Con flotte on-demand, CodeBuild fornisce elaborazione per le tue build. Le macchine vengono distrutte al termine della costruzione. Le flotte on-demand sono completamente gestite e includono funzionalità di scalabilità automatica per gestire i picchi di domanda.

 Note

Le flotte on demand non supportano macOS.

CodeBuild offre anche flotte a capacità riservata che contengono istanze alimentate da Amazon EC2 gestite da CodeBuild. Con le flotte a capacità riservata, puoi configurare un set di istanze dedicate per il tuo ambiente di costruzione. Queste macchine rimangono inattive, pronte per l'elaborazione immediata di build o test e riducono la durata di costruzione. Con flotte a capacità riservata, le macchine sono sempre in funzione e continueranno a sostenere costi finché verranno rifornite.

 Important

Indipendentemente dalla durata di esecuzione di un'istanza, le flotte a capacità riservata sono soggette a un costo iniziale per istanza, dopodiché potrebbero esserci costi aggiuntivi associati. Per ulteriori informazioni, consulta <https://aws.amazon.com/codebuild/pricing/>.

Argomenti

- [Crea un parco veicoli a capacità riservata](#)
- [Best practice](#)
- [Posso condividere una flotta di capacità riservata tra più progetti? CodeBuild](#)
- [Come funziona il calcolo basato sugli attributi?](#)
- [Posso specificare manualmente un'istanza Amazon EC2 per la mia flotta?](#)
- [Quali regioni supportano flotte a capacità riservata?](#)
- [Come posso configurare una flotta macOS a capacità riservata?](#)
- [Come posso configurare un'Amazon Machine Image \(AMI\) personalizzata per un parco macchine a capacità riservata?](#)
- [Limitazioni delle flotte a capacità riservata](#)

- [Proprietà del parco veicoli a capacità riservata](#)
- [Esempi di capacità riservata con AWS CodeBuild](#)

Crea un parco veicoli a capacità riservata

Utilizza le seguenti istruzioni per creare un parco veicoli a capacità riservata.

Per creare un parco veicoli con capacità riservata

1. Accedi Console di gestione AWS e apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel pannello di navigazione, scegli Compute fleets, quindi scegli Create Fleet.
3. Nel campo di testo Compute fleet name, inserisci un nome per la tua flotta.
4. Dal menu a discesa Sistema operativo, scegli il sistema operativo.
5. Dal menu a discesa Architettura, scegli l'architettura.
6. (Facoltativo) Seleziona Usa la modalità di esecuzione dell'istanza: facoltativo per l'esecuzione diretta su un'istanza Amazon EC2 anziché su un contenitore Docker. Quindi scegli una versione principale e una versione secondaria.
7. (Facoltativo) In Configurazione aggiuntiva, procedi come segue:
 - Seleziona Configura VPC: opzionale per connettere la tua flotta a un VPC per accedere a risorse private durante l'utilizzo.
 - Dal menu a discesa VPC, seleziona un VPC a cui accederà la tua flotta. CodeBuild
 - Dal menu a discesa Sottoreti, seleziona le sottoreti da CodeBuild utilizzare per configurare la configurazione del VPC.
 - Dal menu a discesa Gruppi di sicurezza, seleziona i gruppi di sicurezza da CodeBuild utilizzare per lavorare con il tuo VPC.
 - Nel campo Fleet Service Role, scegli un ruolo di servizio esistente.

Note

Assicurati che il tuo ruolo nella flotta disponga delle autorizzazioni necessarie. Per ulteriori informazioni, consulta [Consenti a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service](#).

- Se hai scelto il sistema operativo Amazon Linux, seleziona Definisci configurazioni proxy - opzionale per applicare il controllo dell'accesso alla rete alle tue istanze a capacità riservata.
- Per il comportamento predefinito, scegli di consentire o negare il traffico in uscita verso tutte le destinazioni per impostazione predefinita.
- Per le regole proxy, scegli Aggiungi regola proxy per specificare i domini di destinazione o IPs per consentire o negare il controllo dell'accesso alla rete.
- Seleziona Configura AMI personalizzata: facoltativo per utilizzare un'Amazon Machine Image (AMI) personalizzata.
- Dal menu a discesa AMI, seleziona un'Amazon Machine Image (AMI) per la tua flotta.
- Nel campo Fleet Service Role, scegli un ruolo di servizio esistente.

 Note

Assicurati che il tuo ruolo nella flotta disponga delle autorizzazioni necessarie. Per ulteriori informazioni, consulta [Consenti a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service](#).

8. Nella configurazione della capacità, dalla modalità di selezione Compute, scegli una delle seguenti opzioni:
- Se scegli Selezione guidata, procedi come segue:
 - Per Compute, scegli il tipo di istanze incluse in questo parco istanze.
 - Nel campo di testo Capacità, inserisci il numero minimo di istanze nel parco istanze.
 - (Facoltativo) In Configurazione aggiuntiva, procedi come segue:
 - Seleziona Configura scalabilità: opzionale per ridimensionare automaticamente la flotta in base a questa configurazione. Dal menu a discesa della modalità Scaling, opzionale, scegli il comportamento quando la domanda supera la capacità della flotta.
 - Se scegli Istanza personalizzata, procedi come segue:
 - Dal menu a discesa del tipo di istanza Compute, seleziona il tipo di istanze incluse in questo parco istanze.
 - Nel campo di testo opzionale Dimensioni aggiuntive del volume EBS, inserisci il volume aggiuntivo rispetto ai 64 GB di spazio su disco forniti.
 - Nel campo di testo Capacità, inserisci il numero minimo di istanze nel parco istanze.
 - (Facoltativo) In Configurazione aggiuntiva, procedi come segue:

- Seleziona Configura scalabilità: opzionale per ridimensionare automaticamente la flotta in base a questa configurazione. Dal menu a discesa della modalità Scaling, opzionale, scegli il comportamento quando la domanda supera la capacità della flotta.

9. Scegli Create Compute Fleet.

10. Dopo aver creato la flotta di elaborazione, crea un nuovo CodeBuild progetto o modificane uno esistente. Da Ambiente, scegli Capacità riservata in Modello di provisioning, quindi scegli la flotta specificata in Fleet name.

Best practice

Quando utilizzi flotte a capacità riservata, ti consigliamo di seguire queste migliori pratiche.

- Ti consigliamo di utilizzare la modalità cache del codice sorgente per migliorare le prestazioni di compilazione memorizzando nella cache il codice sorgente.
- Ti consigliamo di utilizzare la memorizzazione nella cache dei livelli Docker per migliorare le prestazioni di compilazione memorizzando nella cache i livelli Docker esistenti.

Posso condividere una flotta di capacità riservata tra più progetti? CodeBuild

Sì, puoi massimizzare l'utilizzo della capacità di una flotta utilizzandola in più progetti.

Important

Quando si utilizza la funzionalità di capacità riservata, i dati memorizzati nella cache delle istanze del parco istanze, inclusi i file sorgente, i layer Docker e le directory memorizzate nella cache specificate nelle specifiche di costruzione, possono essere accessibili ad altri progetti all'interno dello stesso account. Questa funzionalità è preimpostata e consente ai progetti all'interno dello stesso account di condividere le istanze del parco istanze.

Come funziona il calcolo basato sugli attributi?

Se scegli `ATTRIBUTE_BASED_COMPUTE` come quelli del tuo parco `veicolicomputeType`, puoi specificare gli attributi in un nuovo campo chiamato `computeConfiguration`. Questi attributi

includono vCPUs, memoria, spazio su disco `emachineType`. Questo `machineType` è `GENERAL` o `ONVME`. Dopo aver specificato uno o alcuni degli attributi disponibili, CodeBuild sceglierà un tipo di calcolo tra i tipi di istanza supportati disponibili come finalizzato. `computeConfiguration`

Note

CodeBuild sceglierà l'istanza più economica che soddisfa tutti i requisiti di input. La memoria CPUs, v e lo spazio su disco delle istanze scelte saranno tutti maggiori o uguali ai requisiti di input. È possibile verificare la risoluzione `computeConfiguration` nel parco dati creato o aggiornato.

Se inserisci un messaggio `computeConfiguration` che non è possibile soddisfare CodeBuild, riceverai un'eccezione di convalida. Tieni inoltre presente che il comportamento di sovraccarico della flotta su richiesta verrà sostituito dal comportamento in coda se non è disponibile per la `computeConfiguration` modalità on demand.

Posso specificare manualmente un'istanza Amazon EC2 per la mia flotta?

Sì, puoi inserire direttamente l'istanza Amazon EC2 desiderata nella console selezionando *Istanza personalizzata* o configurando il parametro API, `InstanceType`. Questo campo viene utilizzato nei seguenti casi API: `CreateFleet`, `UpdateFleet`, `CreateProject`, `UpdateProject` e `StartBuild`. Per ulteriori informazioni, consulta [Compute instance type](#).

Quali regioni supportano flotte a capacità riservata?

Le flotte Amazon Linux e Windows con capacità riservata sono supportate nei seguenti paesi Regioni AWS: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Mumbai), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte), Europa (Irlanda) e Sud America (San Paolo). [Per ulteriori informazioni su Regioni AWS dove CodeBuild è disponibile, consulta Servizi per regione.AWS](#)

Le flotte macOS Medium a capacità riservata sono supportate nei seguenti paesi Regioni AWS: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Sydney) ed Europa (Francoforte). Le flotte macOS Large con capacità riservata sono supportate nelle seguenti aree Regioni AWS: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon) e Asia Pacifico (Sydney).

Come posso configurare una flotta macOS a capacità riservata?

Per configurare una flotta macOS a capacità riservata

1. Accedi Console di gestione AWS e apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel pannello di navigazione, scegli Compute fleets, quindi scegli Create Fleet.
3. Nel campo di testo Compute fleet name, inserisci un nome per la tua flotta.
4. Dal menu a discesa Sistema operativo, scegli macOS.
5. Nel campo Calcolo, scegli uno dei seguenti tipi di macchina di elaborazione: Apple M2, 24 GB di memoria, 8 v CPUs o Apple M2, 32 GB di memoria, 12 v. CPUs
6. Nel campo di testo Capacità, inserisci il numero minimo di istanze nel parco istanze.
7. (Facoltativo) Per utilizzare un'immagine personalizzata per la tua flotta, [Come posso configurare un'Amazon Machine Image \(AMI\) personalizzata per un parco macchine a capacità riservata?](#) verifica che la tua Amazon Machine Image (AMI) abbia i prerequisiti richiesti.
8. (Facoltativo) Per configurare un VPC con la tua flotta, in Configurazione aggiuntiva procedi come segue:
 - Dal menu a discesa VPC, opzionale, seleziona un VPC a cui accederà la tua flotta. CodeBuild
 - Dal menu a discesa Sottoreti, seleziona le sottoreti da CodeBuild utilizzare per configurare la configurazione del VPC.
 - Dal menu a discesa Gruppi di sicurezza, seleziona i gruppi di sicurezza da CodeBuild utilizzare per lavorare con il tuo VPC.
 - Nel campo Fleet service role, scegli un ruolo di servizio esistente.

Note

Assicurati che il tuo ruolo nella flotta disponga delle autorizzazioni necessarie. Per ulteriori informazioni, consulta [Consenti a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service](#).

9. Scegli Create Compute Fleet e attendi l'avvio dell'istanza del parco istanze. Una volta avviata, la capacità sarà *n*/*n*, *n* dov'è la capacità fornita.

10. Dopo il lancio della flotta di elaborazione, crea un nuovo CodeBuild progetto o modificane uno esistente. Da Ambiente, scegli Capacità riservata in Modello di provisioning, quindi scegli la flotta specificata in Fleet name.

Come posso configurare un'Amazon Machine Image (AMI) personalizzata per un parco macchine a capacità riservata?

Per configurare un'Amazon Machine Image (AMI) personalizzata per un parco macchine a capacità riservata

1. Accedi Console di gestione AWS e apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel pannello di navigazione, scegli Compute fleets, quindi scegli Create Fleet.
3. Nel campo di testo Compute fleet name, inserisci un nome per la tua flotta.
4. Scegli un'immagine personalizzata per la tua flotta e assicurati che la tua Amazon Machine Image (AMI) soddisfi i seguenti prerequisiti:
 - Se il tipo di ambiente èMAC_ARM, assicurati che l'architettura AMI sia a 64 bitMac-Arm.
 - Se il tipo di ambiente èLINUX_EC2, assicurati che l'architettura AMI sia a 64 bitx86.
 - Se il tipo di ambiente èARM_EC2, assicurati che l'architettura AMI sia a 64 bitArm.
 - Se il tipo di ambiente èWINDOWS_EC2, assicurati che l'architettura AMI sia a 64 bitx86.
 - L'AMI consente il CodeBuild servizio Organization ARN. Per un elenco delle organizzazioni ARNs, vedere[Amazon Machine Images \(AMI\)](#).
 - Se l'AMI è crittografato con una AWS KMS chiave, la AWS KMS chiave deve consentire anche l'ID dell'organizzazione del CodeBuild servizio. Per un elenco delle organizzazioni IDs, vedere[Amazon Machine Images \(AMI\)](#). Per ulteriori informazioni sulle AWS KMS chiavi, consulta [Consentire alle organizzazioni e OUs di utilizzare una chiave KMS nella Guida](#) per l'utente di Amazon EC2. Per autorizzare CodeBuild l'organizzazione a utilizzare una chiave KMS, aggiungi la seguente dichiarazione alla politica chiave:

```
{
  "Sid": "Allow access for organization root",
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
    "kms:Describe*",
```

```
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

- Nel campo Fleet service role, concedi le seguenti autorizzazioni Amazon EC2:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots"
      ],
      "Resource": "*"
    }
  ]
}
```

Limitazioni delle flotte a capacità riservata

Esistono alcuni casi d'uso che le flotte a capacità riservata non sono supportate e, se hanno un impatto su di te, utilizza invece flotte su richiesta:

- Le flotte a capacità riservata non supportano le metriche di utilizzo delle build.
- Le flotte macOS a capacità riservata non supportano le sessioni di debug.

Per ulteriori informazioni su limiti e quote, consulta [Flotte di calcolo](#)

Proprietà del parco veicoli a capacità riservata

Un parco veicoli con capacità riservata contiene le seguenti proprietà. Per ulteriori informazioni sulle flotte a capacità riservata, vedere [Run si basa su flotte a capacità riservata](#).

Sistema operativo

Il sistema operativo Sono disponibili i seguenti sistemi operativi:

- Amazon Linux
- macOS
- Windows Server 2019
- Windows Server 2022

Architecture

L'architettura del processore. Sono disponibili le seguenti architetture:

- x86_64
- Arm64

Tipo di ambiente

I tipi di ambiente disponibili quando è selezionato Amazon Linux. Sono disponibili i seguenti tipi di ambiente:

- Linux EC2
- GPU Linux

Tipo di istanza di calcolo

Le configurazioni di calcolo per le istanze della flotta.

Selezione guidata

Specificate diversi tipi di elaborazione selezionando le impostazioni di vCPU, memoria e spazio su disco. Per informazioni sulla disponibilità dei tipi di calcolo per regione, consulta.

[Informazioni sui tipi di ambienti con capacità riservata](#)

Istanza personalizzata

Specificate manualmente il tipo di istanza desiderato.

Capacity

Il numero iniziale di macchine assegnate alla flotta, che definisce il numero di build che possono essere eseguite in parallelo.

Comportamento di overflow

Definisce il comportamento quando il numero di build supera la capacità della flotta.

Su richiesta

Le build Overflow vengono eseguite su richiesta. CodeBuild

Note

Se scegli di impostare il comportamento di overflow su richiesta durante la creazione di una flotta connessa a VPC, assicurati di aggiungere le autorizzazioni VPC richieste al tuo ruolo di project service. Per ulteriori informazioni, consulta [Esempio di dichiarazione politica per consentire CodeBuild l'accesso ai AWS servizi necessari per creare un'interfaccia di rete VPC](#).

Important

Se scegli di impostare il comportamento di overflow su richiesta, tieni presente che le build overflow verranno fatturate separatamente, in modo simile a Amazon EC2 on-demand. Per ulteriori informazioni, consulta <https://aws.amazon.com/codebuild/pricing/>.

Coda

Le esecuzioni di compilazione vengono messe in coda fino a quando una macchina non è disponibile. Ciò limita i costi aggiuntivi perché non vengono allocate macchine aggiuntive.

Immagini di macchine Amazon (AMI)

Le proprietà di Amazon Machine Image (AMI) per la tua flotta. Le seguenti proprietà sono supportate da CodeBuild:

Regioni AWS	Organizzazione ARN	ID organizzazione
us-east-1	arn:aws:organizations::851725618577:organization/o-c6wcu152r1	o-c6wcu152r1
us-east-2	arn:aws:organizations::992382780434:organization/o-seufr2suvq	o-seufr2suvq
us-west-2	arn:aws:organizations::381491982620:organization/o-0412o99a4r	o-0412o99a4r
ap-northeast-1	arn:aws:organizations::891376993293:organization/o-b6k3sjqavm	o-b6k3sjqavm
ap-south-1	arn:aws:organizations::891376924779:organization/o-krtah1lkeg	o-krtah1lkeg
ap-southeast-1	arn:aws:organizations::654654522137:organization/o-mcn8uvc3tp	o-mcn8uvc3tp
ap-southeast-2	arn:aws:organizations::767398067170:organization/o-6crt0f6bu4	o-6crt0f6bu4

Regioni AWS	Organizzazione ARN	ID organizzazione
eu-central-1	arn:aws:organizations::590183817084:organization/o-lb2lne3te6	o-lb2lne3te6
eu-west-1	arn:aws:organizations::891376938588:organization/o-ullrrg5qf0	o-ullrrg5qf0
sa-east-1	arn:aws:organizations::533267309133:organization/o-db63c45ozw	o-db63c45ozw

Configurazione aggiuntiva

VPC - opzionale

Il VPC a cui accederà la tua CodeBuild flotta. Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Note

Se viene specificata un'override della flotta quando si chiama l' StartBuild API, CodeBuild ignorerà la configurazione VPC del progetto.

Sottoreti

Le sottoreti VPC CodeBuild utilizzate per configurare la configurazione VPC. Tieni presente che le flotte a capacità riservata supportano solo una sottorete in una singola zona di disponibilità. Inoltre, assicurati che le tue sottoreti includano un gateway NAT.

Gruppi di sicurezza

I gruppi di sicurezza VPC CodeBuild utilizzati con il tuo VPC. Assicurati che i tuoi gruppi di sicurezza consentano le connessioni in uscita.

Ruolo del Fleet Service

Definisce il ruolo di servizio per la tua flotta a partire da un ruolo di servizio esistente nel tuo account.

Definire le configurazioni del proxy (opzionale)

Configurazioni proxy che applicano il controllo dell'accesso alla rete alle istanze con capacità riservata. Per ulteriori informazioni, consulta [Utilizzo AWS CodeBuild con un server proxy gestito](#).

Note

Le configurazioni proxy non supportano VPC, Windows o macOS.

Comportamento predefinito

Definisce il comportamento del traffico in uscita.

Abilita

Per impostazione predefinita, consente il traffico in uscita verso tutte le destinazioni.

Rifiuta

Per impostazione predefinita, nega il traffico in uscita verso tutte le destinazioni.

Regole proxy

Specifica i domini di destinazione o IPs a cui consentire o negare il controllo dell'accesso alla rete.

Esempi di capacità riservata con AWS CodeBuild

Questi esempi possono essere usati per sperimentare flotte a capacità riservata in CodeBuild

Argomenti

- [Memorizzazione nella cache con campione di capacità riservata](#)

Memorizzazione nella cache con campione di capacità riservata

Una cache memorizza parti riutilizzabili dell'ambiente di compilazione e le utilizza su più compilazioni. Questo esempio ha dimostrato come abilitare la memorizzazione nella cache all'interno del progetto di build utilizzando la capacità riservata. Per ulteriori informazioni, consulta [Creazioni di cache per migliorare le prestazioni](#).

Puoi iniziare specificando una o più modalità cache nelle impostazioni del progetto:

Cache:

Type: LOCAL

Modes:

- LOCAL_CUSTOM_CACHE
- LOCAL_DOCKER_LAYER_CACHE
- LOCAL_SOURCE_CACHE

Note

Assicurati di abilitare la modalità privilegiata per utilizzare la cache di Docker Layer.

Le impostazioni buildspec del tuo progetto dovrebbero essere simili alle seguenti:

```
version: 0.2
phases:
  build:
    commands:
      - echo testing local source cache
      - touch /codebuild/cache/workspace/foobar.txt
      - git checkout -b cached_branch
      - echo testing local docker layer cache
      - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
      - echo testing local custom cache
      - touch foo
      - mkdir bar && ln -s foo bar/foo2
      - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
      - "[ -f foo ] || exit 1"
      - "[ -L bar/foo2 ] || exit 1"
      - "[ -f bar/bar/foo3 ] || exit 1"
      - "[ -f bar/bar/foo4 ] || exit 1"
cache:
```

```
paths:
  - './foo'
  - './bar/**/*'
  - './bar/bar/foo3'
```

Puoi iniziare eseguendo una build con il nuovo progetto per seminare la cache. Una volta completata, dovresti iniziare un'altra build con una buildspec sostitutiva, simile alla seguente:

```
version: 0.2
phases:
  build:
    commands:
      - echo testing local source cache
      - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
      - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
      - echo testing local docker layer cache
      - docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
      - echo testing local custom cache
      - "[ -f foo ] || exit 1"
      - "[ -L bar/foo2 ] || exit 1"
      - "[ -f bar/bar/foo3 ] || exit 1"
      - "[ -f bar/bar/foo4 ] || exit 1"
    cache:
      paths:
        - './foo'
        - './bar/**/*'
        - './bar/bar/foo3'
```

Esegui le build in batch

Puoi utilizzarlo AWS CodeBuild per eseguire build simultanee e coordinate di un progetto con build in batch.

Argomenti

- [Ruolo di sicurezza](#)
- [Tipi di build in batch](#)
- [Modalità report Batch](#)
- [Ulteriori informazioni](#)

Ruolo di sicurezza

Le compilazioni in batch introducono un nuovo ruolo di sicurezza nella configurazione in batch. Questo nuovo ruolo è obbligatorio in quanto CodeBuild deve essere in grado di richiamare le `RetryBuild` azioni `StartBuild``StopBuild`, e per conto dell'utente per eseguire le build come parte di un batch. I clienti devono utilizzare un nuovo ruolo e non lo stesso ruolo che usano nella compilazione, per due motivi:

- Dare al ruolo di compilazione le autorizzazioni `StartBuild`, `StopBuild` e `RetryBuild` consentirebbe a una singola compilazione di avviare più compilazioni tramite `buildspec`.
- CodeBuild le build in batch forniscono restrizioni che limitano il numero di build e i tipi di calcolo che possono essere utilizzati per le build del batch. Se il ruolo di compilazione dispone di queste autorizzazioni, è possibile che le compilazioni stesse possano ignorare queste restrizioni.

Tipi di build in batch

CodeBuild supporta i seguenti tipi di build in batch:

Tipi di build in batch

- [Costruisci un grafico](#)
- [Crea un elenco](#)
- [Costruisci una matrice](#)
- [Crea fanout](#)

Costruisci un grafico

Un grafico di compilazione definisce un insieme di attività che dipendono da altre attività del batch.

L'esempio seguente definisce un grafico di compilazione che crea una catena di dipendenze.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
```

```
- identifier: build2
  buildspec: build2.yml
  env:
    variables:
      BUILD_ID: build2
  depend-on:
    - build1
- identifier: build3
  env:
    variables:
      BUILD_ID: build3
  depend-on:
    - build2
- identifier: build4
  env:
    compute-type: ARM_LAMBDA_1GB
- identifier: build5
  env:
    fleet: fleet_name
```

In questo esempio:

- `build1` viene eseguito per primo perché non ha dipendenze.
- `build2` ha una dipendenza da `build1`, quindi `build2` viene eseguito dopo `build1` il completamento.
- `build3` dipende da `build2`, quindi `build3` viene eseguito dopo il completamento. `build2`

Per ulteriori informazioni sulla sintassi `build graph buildspec`, vedi [batch/build-graph](#)

Crea un elenco

Un elenco di build definisce una serie di attività eseguite in parallelo.

L'esempio seguente definisce un elenco di build. Le build `build2` `build build1` and verranno eseguite in parallelo.

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
```

```
variables:
  BUILD_ID: build1
ignore-failure: false
- identifier: build2
  buildspec: build2.yml
env:
  variables:
    BUILD_ID: build2
  ignore-failure: true
- identifier: build3
env:
  compute-type: ARM_LAMBDA_1GB
- identifier: build4
env:
  fleet: fleet_name
- identifier: build5
env:
  compute-type: GENERAL_LINUX_XLAGRE
```

Per ulteriori informazioni sulla sintassi buildspec della lista di compilazione, consulta. [batch/build-list](#)

Costruisci una matrice

Una matrice di compilazione definisce le attività con diverse configurazioni eseguite in parallelo. CodeBuild crea una build separata per ogni possibile combinazione di configurazione.

L'esempio seguente mostra una matrice di compilazione con due file buildspec e tre valori per una variabile di ambiente.

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
```

- VALUE2
- VALUE3

In questo esempio, CodeBuild crea sei build:

- `matrix1.yml` con `$MY_VAR=VALUE1`
- `matrix1.yml` con `$MY_VAR=VALUE2`
- `matrix1.yml` con `$MY_VAR=VALUE3`
- `matrix2.yml` con `$MY_VAR=VALUE1`
- `matrix2.yml` con `$MY_VAR=VALUE2`
- `matrix2.yml` con `$MY_VAR=VALUE3`

Ogni build avrà le seguenti impostazioni:

- `ignore-failure` impostato su `false`
- `env/type` impostato su `LINUX_CONTAINER`
- `env/image` impostato su `aws/codebuild/amazonlinux-x86_64-standard:4.0`
- `env/privileged-mode` impostato su `true`

Queste build vengono eseguite in parallelo.

Per ulteriori informazioni sulla sintassi `buildspec` della matrice di compilazione, vedere [batch/build-matrix](#)

Crea fanout

Un fanout di compilazione definisce un'attività che verrà suddivisa in più build nel batch. Può essere usato per eseguire test in parallelo. CodeBuild crea una build separata per ogni frammento di casi di test in base al valore impostato nel `parallelism` campo.

L'esempio seguente definisce un build fanout che crea cinque build che vengono eseguite in parallelo.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
```

```
parallelism: 5
ignore-failure: false

phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'equal-distribution'
```

In questo esempio, supponendo che ci siano 100 test da eseguire, CodeBuild crea cinque build che eseguono ciascuna 20 test in parallelo.

Per ulteriori informazioni sulla sintassi build graph buildspec, consulta [batch/build-fanout](#)

Modalità report Batch

Se il provider di origine del progetto è Bitbucket o GitHub Enterprise e il progetto è configurato per riportare gli stati di compilazione al provider di origine, puoi selezionare il modo in cui desideri che gli stati della compilazione in batch vengano inviati al provider di origine. GitHub Puoi scegliere di inviare gli stati come un unico rapporto di stato aggregato per il batch o di riportare singolarmente lo stato di ogni build del batch.

Per ulteriori informazioni, consulta i seguenti argomenti:

- [Configurazione Batch \(creazione\)](#)
- [Configurazione in batch \(aggiornamento\)](#)

Ulteriori informazioni

Per ulteriori informazioni, consulta i seguenti argomenti:

- [Riferimento buildspec per la compilazione in batch](#)
- [Configurazione Batch](#)

- [Esegui una build in batch \(AWS CLI\)](#)
- [Interrompi gli accumuli in batch AWS CodeBuild](#)

Esegui test paralleli in build in batch

È possibile utilizzare AWS CodeBuild per eseguire test paralleli nelle build in batch. L'esecuzione parallela dei test è un approccio di test in cui più casi di test vengono eseguiti contemporaneamente su ambienti, macchine o browser diversi, anziché essere eseguiti in sequenza. Questo approccio può ridurre in modo significativo il tempo complessivo di esecuzione dei test e migliorare l'efficienza dei test. Inoltre CodeBuild, puoi suddividere i test in più ambienti ed eseguirli contemporaneamente.

I principali vantaggi dell'esecuzione parallela dei test includono:

1. Tempo di esecuzione ridotto: i test che richiederebbero ore in sequenza possono essere completati in pochi minuti.
2. Migliore utilizzo delle risorse: utilizza in modo efficiente le risorse di elaborazione disponibili.
3. Feedback anticipato: un completamento più rapido del test significa un feedback più rapido per gli sviluppatori.
4. Conveniente: consente di risparmiare tempo e costi di elaborazione a lungo termine.

Quando si implementa l'esecuzione di test paralleli, vengono comunemente considerati due approcci principali: ambienti separati e multithreading. Sebbene entrambi i metodi mirino a ottenere l'esecuzione simultanea dei test, differiscono in modo significativo nella loro implementazione ed efficacia. Ambienti separati creano istanze isolate in cui ogni suite di test viene eseguita in modo indipendente, mentre il multithreading esegue più test contemporaneamente all'interno dello stesso spazio di processo utilizzando thread diversi.

I vantaggi principali degli ambienti separati rispetto al multithreading includono:

1. Isolamento: ogni test viene eseguito in un ambiente completamente isolato, evitando interferenze tra i test.
2. Conflitti di risorse: nessuna concorrenza per le risorse condivise, cosa che spesso si verifica nel multithreading.
3. Stabilità: meno incline alle condizioni di gara e ai problemi di sincronizzazione.
4. Debug più semplice: quando i test falliscono, è più semplice identificare la causa poiché ogni ambiente è indipendente.

5. Gestione dello stato: gestisci facilmente i problemi di stato condivisi che affliggono i test multithread.
6. Migliore scalabilità: può aggiungere facilmente più ambienti senza complessità.

Argomenti

- [Support in AWS CodeBuild](#)
- [Abilita l'esecuzione di test paralleli nelle build in batch](#)
- [Usa il comando codebuild-tests-run CLI](#)
- [Usa il comando codebuild-glob-search CLI](#)
- [Informazioni sulla suddivisione dei test](#)
- [Unisci automaticamente i report di build individuali](#)
- [Esempio di esecuzione parallela di test per vari framework di test](#)

Support in AWS CodeBuild

AWS CodeBuild fornisce un solido supporto per l'esecuzione di test paralleli tramite la sua funzionalità di creazione in batch, progettata specificamente per sfruttare l'esecuzione in ambienti separati. Questa implementazione si allinea perfettamente con i vantaggi degli ambienti di test isolati.

Creazione in batch con distribuzione dei test

CodeBuild la funzionalità di compilazione in batch consente la creazione di più ambienti di compilazione che vengono eseguiti contemporaneamente. Ogni ambiente funziona come un'unità completamente isolata, con risorse di elaborazione, ambiente di runtime e dipendenze propri. Tramite la configurazione di build in batch, puoi specificare quanti ambienti paralleli sono necessari e come i test devono essere distribuiti su di essi.

CLI di partizionamento dei test

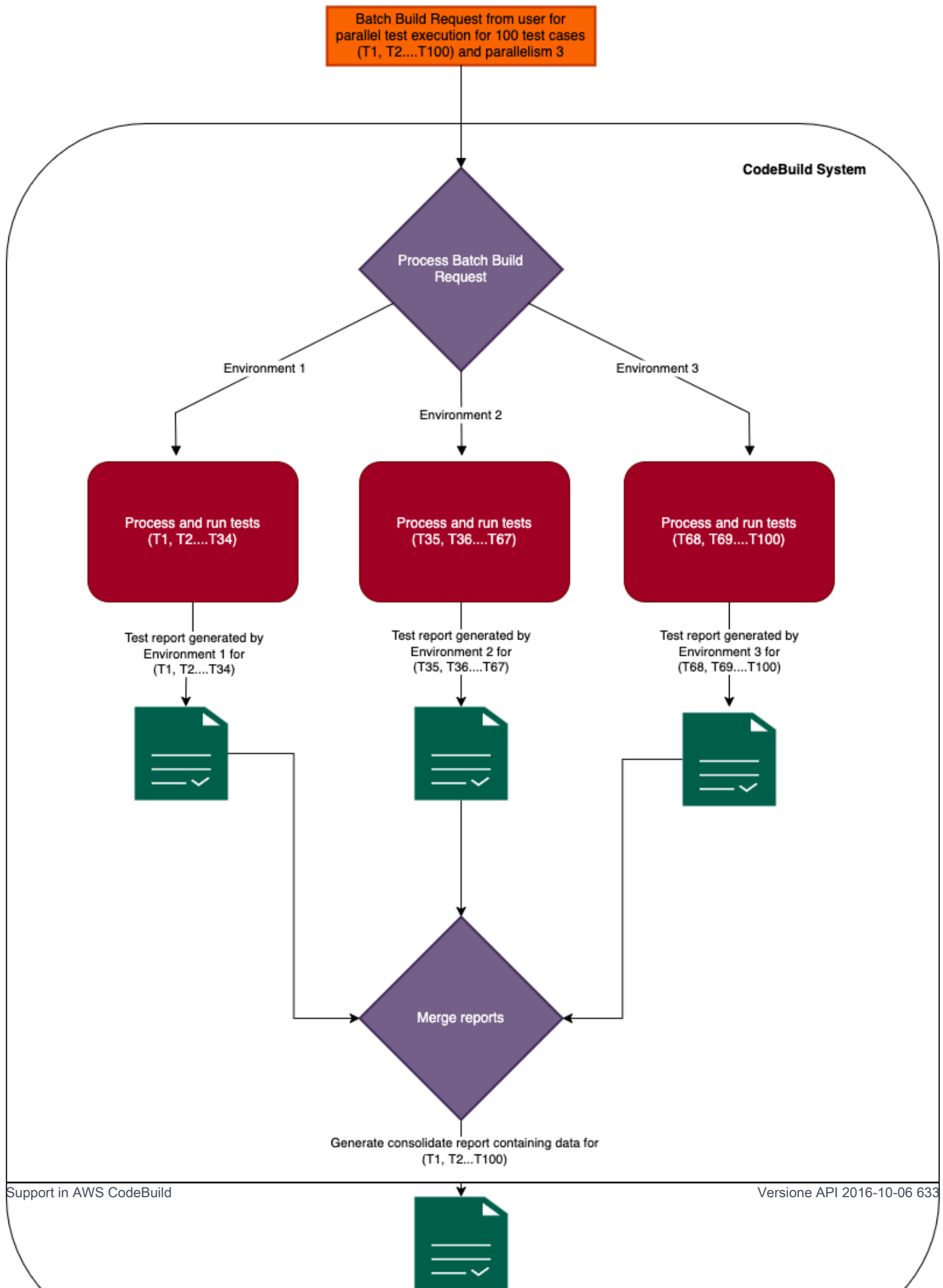
CodeBuild include un meccanismo di distribuzione dei test integrato tramite il suo strumento CLI `codebuild-tests-run`, che divide automaticamente i test in diversi ambienti.

Aggregazione dei report

Uno dei principali punti di forza dell'implementazione è la sua capacità CodeBuild di gestire l'aggregazione dei risultati dei test senza problemi. Mentre i test vengono eseguiti in ambienti separati, raccoglie e combina CodeBuild automaticamente i report di test di ciascun ambiente in un rapporto di test unificato a livello di creazione del batch. Questo consolidamento fornisce una

visione completa dei risultati dei test mantenendo i vantaggi in termini di efficienza dell'esecuzione parallela.

Di seguito è riportato il diagramma che spiega il concetto completo di esecuzione di test paralleli in AWS CodeBuild.



Abilita l'esecuzione di test paralleli nelle build in batch

Per eseguire i test in parallelo, aggiorna il file batch build buildspec per includere il campo `build-fanout` e il numero di build parallele per dividere la suite di test nel campo come mostrato di seguito. Il `parallelism` campo specifica quanti esecutori indipendenti sono configurati per eseguire la suite di test.

Per eseguire i test in più ambienti di esecuzione parallela, imposta il `parallelism` campo su un valore maggiore di zero. Nell'esempio seguente, `parallelism` è impostato su cinque, il che significa che CodeBuild avvia cinque build identiche che eseguono una parte della suite di test in parallelo.

Puoi usare il comando [codebuild-tests-run](#) CLI per dividere ed eseguire i test. I file di test verranno suddivisi e una parte dei test verrà eseguita in ogni build. Ciò riduce il tempo complessivo impiegato per eseguire l'intera suite di test. Nell'esempio seguente, i test verranno suddivisi in cinque e i punti di divisione verranno calcolati in base al nome dei test.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install jest-junit --save-dev
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - |
        codebuild-tests-run \
          --test-command 'npx jest --runInBand --coverage' \
          --files-search "codebuild-glob-search '**/_tests_/**/*test.js'" \
          --sharding-strategy 'equal-distribution'

  post_build:
    commands:
```

```
- codebuild-glob-search '**/*.xml'
- echo "Running post-build steps..."
- echo "Build completed on `date`"

reports:
  test-reports:
    files:
      - '**/junit.xml'
    base-directory: .
    discard-paths: yes
    file-format: JUNITXML
```

Se i report sono configurati per la build `build-fanout`, i report di test vengono generati separatamente per ogni build, che possono essere visualizzati nella scheda Report delle build corrispondenti nella console. AWS CodeBuild

Per ulteriori informazioni su come eseguire test paralleli in batch, vedere [Esempio di esecuzione parallela di test per vari framework di test](#).

Usa il comando **codebuild-tests-run** CLI

AWS CodeBuild fornisce una CLI che prenderà come input il comando di test e la posizione del file di test. La CLI con questi input suddividerà i test in un numero di frammenti come specificato nel `parallelism` campo in base ai nomi dei file di test. L'assegnazione dei file di test agli shard viene decisa dalla strategia di sharding.

```
codebuild-tests-run \
  --files-search "codebuild-glob-search '**/__tests__/*.js'" \
  --test-command 'npx jest --runInBand --coverage' \
  --sharding-strategy 'equal-distribution'
```

La tabella seguente descrive i campi per il comando `codebuild-tests-run` CLI.

Nome del campo	Tipo	Obbligatorio o facoltativo	Definizione
<code>test-command</code>	Stringa	Richiesto	Questo comando viene utilizzato per eseguire i test.

Nome del campo	Tipo	Obbligatorio o facoltativo	Definizione
<code>files-search</code>	Stringa	Richiesto	Questo comando fornisce un elenco di file di test. Puoi utilizzare il comando codebuild-glob-search CLI AWS CodeBuild fornito o qualsiasi altro strumento di ricerca di file a tua scelta.  Note Assicurati che il <code>files-search</code> comando restituisca i nomi dei file, ciascuno separato da una nuova riga.

Nome del campo	Tipo	Obbligatorio o facoltativo	Definizione
sharding-strategy	Enum	Facoltativo	Valori validi: <code>equal-distribution</code> (valore predefinito), <code>stability</code> • <code>equal-distribution</code> : suddivide i file di test in modo uniforme in base ai nomi dei file di test. • <code>stability</code> : suddivide i file di test utilizzando un hashing coerente dei nomi dei file. Per ulteriori informazioni, consulta Informazioni sulla suddivisione dei test.

La `codebuild-tests-run` CLI funziona innanzitutto per identificare l'elenco dei file di test utilizzando il comando fornito nel `files-search` parametro. Quindi determina un sottoinsieme di file di test designati per lo shard (ambiente) corrente utilizzando la strategia di sharding specificata. Infine, questo sottoinsieme di file di test viene formattato in un elenco separato da spazi e aggiunto alla fine del comando fornito nel parametro prima di essere eseguito. `test-command`

Per i framework di test che non accettano elenchi separati da spazi, la `codebuild-tests-run` CLI fornisce un'alternativa flessibile tramite la variabile di ambiente.

CODEBUILD_CURRENT_SHARD_FILES Questa variabile contiene un elenco separato da nuova riga di percorsi di file di test designati per lo shard di build corrente. Sfruttando questa variabile di ambiente, puoi adattarti facilmente a vari requisiti del framework di test, soddisfacendo quelli che

prevedono formati di input diversi dagli elenchi separati da spazi. Inoltre, puoi anche formattare i nomi dei file di test in base alle esigenze del framework di test. Di seguito è riportato un esempio dell'utilizzo di `CODEBUILD_CURRENT_SHARD_FILES` su Linux con il framework Django. Qui `CODEBUILD_CURRENT_SHARD_FILES` viene utilizzato per ottenere i percorsi dei file con notazione a punti supportati da Django:

```
codebuild-tests-run \
  -files-search "codebuild-glob-search '/tests/test.py'" \
  -test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
-E "s/\//_/g; s/\.py$/; s/_/./g")' \
  -sharding-strategy 'equal-distribution'
```

Note

Nota che la variabile di `CODEBUILD_CURRENT_SHARD_FILES` ambiente può essere utilizzata solo all'interno dell'ambito della `codebuild-tests-run` CLI.

Inoltre, se stai usando `CODEBUILD_CURRENT_SHARD_FILES` inside `test-command`, inserisci `CODEBUILD_CURRENT_SHARD_FILES` tra virgolette doppie come mostrato nell'esempio precedente.

Usa il comando **codebuild-glob-search** CLI

AWS CodeBuild fornisce uno strumento CLI integrato chiamato `codebuild-glob-search` che consente di cercare file nella directory di lavoro in base a uno o più modelli di glob. Questo strumento può essere particolarmente utile quando si desidera eseguire test su file o directory specifici all'interno del progetto.

Utilizzo

La `codebuild-glob-search` CLI ha la seguente sintassi di utilizzo:

```
codebuild-glob-search <glob_pattern1> [<glob_pattern2> ...]
```

- **<glob_pattern1><glob_pattern2>**, ecc.: Uno o più pattern globulari da confrontare con i file nella directory di lavoro.
- *****: Corrisponde a qualsiasi sequenza di caratteri (esclusi i separatori di percorso).
- ******: corrisponde a qualsiasi sequenza di caratteri (inclusi i separatori di percorso).

Note

Assicurati che la stringa glob contenga virgolette. Per verificare i risultati del pattern-matching, usa il comando. `echo`

```
version: 0.2

phases:
  build:
    commands:
      - echo $(codebuild-glob-search '**/__tests__/*.js')
      - codebuild-glob-search '**/__tests__/*.js' | xargs -n 1 echo
```

Output

La CLI produrrà un elenco separato da nuove righe di percorsi di file che corrispondono ai modelli di glob forniti. I percorsi dei file restituiti saranno relativi alla directory di lavoro.

Se non viene trovato alcun file corrispondente ai modelli forniti, la CLI emetterà un messaggio che indica che non è stato trovato alcun file.

Tieni presente che le directory trovate a causa di un determinato modello verranno escluse dai risultati della ricerca.

Esempio

Se vuoi cercare solo i file all'interno della directory `tests` e delle sue sottodirectory con `.js` estensione, puoi usare il seguente comando con la `codebuild-glob-search` CLI:

```
codebuild-glob-search '**/__tests__/*.js'
```

Questo comando cercherà tutti i file con `.js` estensione all'interno della `__tests__` directory e delle relative sottodirectory, come indicato dal modello.

Informazioni sulla suddivisione dei test

AWS CodeBuild consente di parallelizzare l'esecuzione della suite di test su più istanze di calcolo, riducendo il tempo complessivo di esecuzione del test. Questa funzionalità è abilitata tramite la

configurazione batch nelle impostazioni del CodeBuild progetto e l'`codebuild-tests-run` utilità nel file `buildspec`.

I test vengono suddivisi in base alla strategia di sharding specificata. CodeBuild fornisce due strategie di sharding come specificato di seguito:

Distribuzione equa

La strategia `equal-distribution` di sharding divide i test in build parallele in base all'ordine alfabetico dei nomi dei file di test. Questo approccio ordina innanzitutto i file di test e poi utilizza un metodo basato su blocchi per distribuirli, assicurando che file simili vengano raggruppati insieme per i test. È consigliato quando si ha a che fare con un insieme relativamente piccolo di file di test. Sebbene questo metodo miri ad allocare un numero approssimativamente uguale di file a ogni shard, con una differenza massima di uno, non garantisce la stabilità. Quando i file di test vengono aggiunti o rimossi nelle build successive, la distribuzione dei file esistenti può cambiare, causando potenzialmente la riassegnazione tra gli shard.

Stabilità

La strategia di `stability sharding` utilizza un algoritmo di hashing coerente per suddividere i test tra gli shard, garantendo che la distribuzione dei file rimanga stabile. Quando vengono aggiunti o rimossi nuovi file, questo approccio garantisce che le file-to-shard assegnazioni esistenti rimangano sostanzialmente invariate. Per suite di test di grandi dimensioni, si consiglia di utilizzare l'opzione di stabilità per distribuire uniformemente i test tra gli shard. Questo meccanismo mira a fornire una distribuzione quasi uguale, garantendo che ogni shard riceva un numero simile di file, con una varianza minima. Sebbene la strategia di stabilità non garantisca una distribuzione equa ideale, offre una distribuzione quasi uguale che mantiene la coerenza nelle assegnazioni dei file tra le build, anche quando i file vengono aggiunti o rimossi.

Per abilitare la suddivisione dei test, è necessario configurare la sezione batch nelle impostazioni del CodeBuild progetto, specificando il livello desiderato `parallelism` e altri parametri pertinenti. Inoltre, è necessario includere l'`codebuild-tests-run` utilità nel file `buildspec`, insieme ai comandi di test e al metodo di suddivisione appropriati.

Unisci automaticamente i report di build individuali

Nelle build in batch fanout, AWS CodeBuild supporta l'unione automatica dei singoli report di build in un report consolidato a livello di batch. Questa funzionalità offre una visione completa dei risultati dei test e della copertura del codice in tutte le build all'interno di un batch.

Come funziona

Quando si eseguono build fanout in batch, ogni singola build genera report di [test](#). CodeBuild quindi consolida automaticamente report identici provenienti da build diverse in un report unificato, che viene allegato alla build batch. Questi report consolidati sono facilmente accessibili tramite il `reportArns` campo dell' [BatchGetBuildBatches](#) API e possono essere visualizzati anche nella scheda Report della console. Questa funzionalità di fusione si estende anche ai report rilevati automaticamente.

I report consolidati vengono creati in [gruppi di report](#) specificati nelle specifiche della build o scoperti automaticamente da CodeBuild. Puoi analizzare le tendenze dei report uniti direttamente in questi gruppi di report, fornendo informazioni preziose sulle metriche complessive, sulle prestazioni di costruzione e sulle metriche di qualità nelle build storiche dello stesso progetto in batch di compilazione.

Per ogni singola build all'interno del batch, crea CodeBuild automaticamente gruppi di report separati. Questi seguono una convenzione di denominazione specifica, che combina il nome del gruppo di report creato in batch con un suffisso di `BuildFanoutShard<shard_number>`, dove `shard_number` rappresenta il numero dello shard in cui viene creato il gruppo di report. Questa organizzazione consente di tracciare e analizzare le tendenze sia a livello di compilazione consolidato che individuale, offrendo flessibilità nel modo in cui monitorate e valutate i relativi processi di compilazione.

Il rapporto di creazione in batch segue la stessa struttura dei report di compilazione [individuali](#). I seguenti campi chiave nella scheda Report sono specifici per i report creati in batch:

Stato del report relativo alla creazione in Batch

Lo stato dei report di creazione in batch segue regole specifiche a seconda del tipo di report:

- Rapporti di test:
 - Riuscito: lo stato viene impostato su Riuscito quando tutti i singoli report di compilazione hanno avuto esito positivo.
 - Non riuscito: lo stato è impostato su Non riuscito se un singolo report di compilazione non è riuscito.
 - Incompleto: lo stato viene contrassegnato come incompleto se un singolo rapporto di compilazione è mancante o presenta uno stato incompleto.
- Rapporti sulla copertura del codice:

- **Completo:** lo stato viene impostato per essere completato quando tutti i report di compilazione individuali sono completi.
- **Non riuscito:** lo stato è impostato su Non riuscito se un singolo report di compilazione non è riuscito.
- **Incompleto:** lo stato viene contrassegnato come incompleto se un singolo rapporto di compilazione è mancante o presenta uno stato incompleto.

Riepilogo del test

Il rapporto di test unito consolida i seguenti campi di tutti i singoli report di build:

- **duration-in-nano-seconds:** Durata massima del test in nanosecondi tra tutti i singoli report di build.
- **totale:** il conteggio combinato di tutti i casi di test, sommando il numero totale di test di ciascuna build.
- **conteggi degli stati:** fornisce una visualizzazione consolidata degli stati dei test, ad esempio superati, non riusciti o ignorati, calcolati aggregando il conteggio di ogni tipo di stato in tutte le singole build.

Riepilogo della copertura del codice

Il rapporto sulla copertura del codice unito combina i campi di tutte le singole build utilizzando i seguenti calcoli:

- **filiali coperte:** somma di tutte le filiali coperte dai singoli report.
- **succursali perse:** somma di tutte le filiali mancanti riportate nelle segnalazioni individuali.
- **branch-coverage-percentage:** $(\text{Total covered branches} / \text{Total branches}) * 100$
- **righe coperte:** somma di tutte le righe coperte dai singoli report.
- **righe mancanti:** somma di tutte le righe mancanti dei singoli report.
- **lines-coverage-percentage:** $(\text{Total covered lines} / \text{Total lines}) * 100$

ID di esecuzione

L'ARN di compilazione in batch.

Casi di test

Il report unito contiene un elenco consolidato di tutti i casi di test relativi alle singole build, accessibile sia tramite l'[DescribeTestCases](#) API che tramite il rapporto sulla build in batch nella console.

Coperture del codice

Il rapporto sulla copertura del codice unito fornisce informazioni consolidate sulla copertura di linee e filiali per ogni file in tutte le singole build, accessibili sia tramite l'[DescribeCodeCoverages](#) API che tramite il rapporto di creazione in batch nella console. Nota: per i file coperti da più file di test distribuiti su diversi shard, il report unito utilizza i seguenti criteri di selezione:

1. La selezione primaria si basa sulla copertura di linea più alta tra gli shard.
2. Se la copertura di linea è uguale su più frammenti, viene selezionato lo shard con la copertura di diramazione più elevata.

Esempio di esecuzione parallela di test per vari framework di test

Puoi usare il comando `codebuild-tests-run` CLI per dividere ed eseguire i test in ambienti di esecuzione parallela. La sezione seguente fornisce `buildspec.yml` esempi per vari framework, illustrando l'utilizzo del comando `codebuild-tests-run`

- Ogni esempio seguente include un `parallelism` livello di cinque, il che significa che verranno creati cinque ambienti di esecuzione identici per suddividere i test. Puoi scegliere un `parallelism` livello adatto al tuo progetto modificando il `parallelism` valore nella `build-fanout` sezione.
- Ogni esempio seguente mostra la configurazione dei test in modo che vengano divisi in base al nome del file di test, che è quello predefinito. Questo distribuisce i test in modo uniforme negli ambienti di esecuzione parallela.

Prima di iniziare, consulta [Esegui test paralleli in build in batch](#) per ulteriori informazioni.

Per un elenco completo delle opzioni quando si utilizza il comando `codebuild-tests-run` CLI, vedere. [Usa il comando `codebuild-tests-run` CLI](#)

Argomenti

- [Configura test paralleli con Django](#)
- [Configura test paralleli con Elixir](#)
- [Configura test paralleli con Go](#)
- [Configurare test paralleli con Java \(Maven\)](#)
- [Configura test paralleli con Javascript \(Jest\)](#)

- [Configura test paralleli con Kotlin](#)
- [Configura test paralleli con PHPUnit](#)
- [Configura test paralleli con Pytest](#)
- [Configura test paralleli con Ruby \(Cucumber\)](#)
- [Configura test paralleli con Ruby \(\) RSpec](#)

Configura test paralleli con Django

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Django su una piattaforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - sudo yum install -y python3 python3-pip
      - python3 -m ensurepip --upgrade
      - python3 -m pip install django
  pre_build:
    commands:
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Django Tests'
      - |
        codebuild-tests-run \
          --test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES"
| sed -E "s/\//__g; s/\.py$//; s/__/./g")' \
          --files-search "codebuild-glob-search '**/tests/*test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo 'Test execution completed'
```

L'esempio precedente mostra l'utilizzo della variabile `CODEBUILD_CURRENT_SHARD_FILES` di ambiente. Qui `CODEBUILD_CURRENT_SHARD_FILES` viene utilizzato per recuperare i percorsi dei file di notazione a punti supportati da Django. Usa le virgolette doppie `CODEBUILD_CURRENT_SHARD_FILES` all'interno come mostrato sopra.

Configura test paralleli con Elixir

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Elixir su una piattaforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Elixir dependencies'
      - sudo apt update
      - sudo DEBIAN_FRONTEND=noninteractive apt install -y elixir
      - elixir --version
      - mix --version
  pre_build:
    commands:
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Elixir Tests'
      - |
        codebuild-tests-run \
          --test-command 'mix test' \
          --files-search "codebuild-glob-search '**/test/**/*_test.exs'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```


Configura test paralleli con Go

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Go su una piattaforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Fetching Go version'
      - go version
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running go Tests'
      - go mod init calculator
      - cd calc
      - |
        codebuild-tests-run \
          --test-command "go test -v calculator.go" \
          --files-search "codebuild-glob-search '**/*test.go'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Nell'esempio precedente, `calculator.go` la funzione contiene semplici funzioni matematiche da testare e tutti i file e i file di test si trovano all'interno della `calc` cartella. `calculator.go`

Configurare test paralleli con Java (Maven)

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Java su una piattaforma Linux:

```
version: 0.2
```

```
batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo "Running mvn test"
      - |
        codebuild-tests-run \
          --test-command 'mvn test -Dtest=$(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
"s|src/test/java/||g; s|.java//g; s|/|.|g; s/ /,/g" | tr "\n" "," | sed "s/,,$//")' \
          --files-search "codebuild-glob-search '**/test/**/*.java'"

  post_build:
    commands:
      - echo "Running post-build steps..."
      - echo "Test execution completed"
```

Nell'esempio riportato, la variabile di ambiente `CODEBUILD_CURRENT_SHARD_FILES` contiene file di test nello shard corrente, separati da nuove righe. Questi file vengono convertiti in un elenco di nomi di classi separati da virgole nel formato accettato dal parametro per Maven. `-Dtest`

Configura test paralleli con Javascript (Jest)

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Javascript su una piattaforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: true
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
```

```
install:
  commands:
    - echo 'Installing Node.js dependencies'
    - apt-get update
    - apt-get install -y nodejs
    - npm install
    - npm install --save-dev jest-junit
pre_build:
  commands:
    - echo 'prebuild'
build:
  commands:
    - echo 'Running JavaScript Tests'
    - |
      codebuild-tests-run \
        --test-command "npx jest" \
        --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
        --sharding-strategy 'stability'
post_build:
  commands:
    - echo 'Test execution completed'
```

Configura test paralleli con Kotlin

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Kotlin su una piattaforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 2
    ignore-failure: false

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - echo 'Installing dependencies'
      - KOTLIN_VERSION="1.8.20" # Replace with your desired version
```

```

- curl -o kotlin-compiler.zip -L "https://github.com/JetBrains/kotlin/releases/
download/v${KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip"
- unzip kotlin-compiler.zip -d /usr/local
- export PATH=$PATH:/usr/local/kotlinc/bin
- kotlin -version
- curl -O https://repo1.maven.org/maven2/org/junit/platform/junit-platform-
console-standalone/1.8.2/junit-platform-console-standalone-1.8.2.jar
pre_build:
  commands:
    - echo 'prebuild'
build:
  commands:
    - echo 'Running Kotlin Tests'
    - |
      codebuild-tests-run \
        --test-command 'kotlinc src/main/kotlin/*.kt $(echo
"$CODEBUILD_CURRENT_SHARD_FILES" | tr "\n" " ") -d classes -cp junit-platform-console-
standalone-1.8.2.jar' \
        --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
    - |
      codebuild-tests-run \
        --test-command '
        java -jar junit-platform-console-standalone-1.8.2.jar --class-path classes
\
          $(for file in $CODEBUILD_CURRENT_SHARD_FILES; do
            class_name=$(basename "$file" .kt)
            echo "--select-class $class_name"
          done)
        ' \
        --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
post_build:
  commands:
    - echo "Test execution completed"

```

Nell'esempio precedente, la `codebuild-tests-run` CLI viene utilizzata due volte. Durante la prima esecuzione, `kotlinc` compila i file. La `CODEBUILD_CURRENT_SHARD_FILES` variabile recupera i file di test assegnati allo shard corrente, che vengono poi convertiti in un elenco separato da spazi. Nella seconda esecuzione, JUnit esegue i test. Ancora una volta, `CODEBUILD_CURRENT_SHARD_FILES` recupera i file di test assegnati allo shard corrente, ma questa volta vengono convertiti in nomi di classe.

Configura test paralleli con PHPUnit

Di seguito è riportato un esempio di un `test buildspec.yml` che mostra l'esecuzione di test paralleli PHPUnit su una piattaforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Install dependencies'
      - composer require --dev phpunit/phpunit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running phpunit Tests'
      - composer dump-autoload
      - |
        codebuild-tests-run \
          --test-command "./vendor/bin/phpunit --debug" \
          --files-search "codebuild-glob-search '**/tests/*Test.php'"
  post_build:
    commands:
      - echo 'Test execution completed'
```

Configura test paralleli con Pytest

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Pytest su una piattaforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
```

```
build-fanout:
  parallelism: 5
  ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - apt-get update
      - apt-get install -y python3 python3-pip
      - pip3 install --upgrade pip
      - pip3 install pytest
  build:
    commands:
      - echo 'Running Python Tests'
      - |
        codebuild-tests-run \
          --test-command 'python -m pytest' \
          --files-search "codebuild-glob-search 'tests/test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Pytest su una piattaforma Windows:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - pip install pytest
  pre_build:
    commands:
      - echo 'prebuild'
```

```
build:
  commands:
    - echo 'Running pytest'
    - |
      & codebuild-tests-run `
        --test-command 'pytest @("$env:CODEBUILD_CURRENT_SHARD_FILES" -split \"`r?`n
        \")' `
        --files-search "codebuild-glob-search '**/test_*.py' '**/*_test.py'" `
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Nell'esempio precedente, la variabile di `CODEBUILD_CURRENT_SHARD_FILES` ambiente viene utilizzata per recuperare i file di test assegnati allo shard corrente e passati come array al comando `pytest`.

Configura test paralleli con Ruby (Cucumber)

Di seguito è riportato un esempio `buildspec.yml` che mostra l'esecuzione di test paralleli con Cucumber su una piattaforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - gem install bundler
      - bundle install
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Cucumber Tests'
      - cucumber --init
```

```
- |
  codebuild-tests-run \
    --test-command "cucumber" \
    --files-search "codebuild-glob-search '**/*.feature'"
post_build:
  commands:
    - echo "Test execution completed"
```

Configura test paralleli con Ruby () RSpec

Di seguito è riportato un esempio di un test buildspec.yml che mostra l'esecuzione di test paralleli RSpec su una piattaforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - apt-get update
      - apt-get install -y ruby ruby-dev build-essential
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo 'Running Ruby Tests'
      - |
        codebuild-tests-run \
          --test-command 'bundle exec rspec' \
          --files-search "codebuild-glob-search 'spec/**/*.spec.rb'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```


Creazioni di cache per migliorare le prestazioni

Durante la creazione del progetto è possibile risparmiare tempo utilizzando una cache. Una cache memorizza parti riutilizzabili dell'ambiente di compilazione e le utilizza su più compilazioni. Il tuo progetto di build può utilizzare uno dei due tipi di caching: Amazon S3 o locale. Se utilizzi una cache locale, devi scegliere una o più delle tre modalità di cache disponibili: cache delle origini, cache di livello Docker e cache personalizzata.

Note

La modalità cache di livello Docker è disponibile solo per l'ambiente Linux. Se scegli questa modalità, devi eseguire la build in modalità privilegiata. CodeBuild ai progetti concessi la modalità privilegiata concede al contenitore l'accesso a tutti i dispositivi. Per ulteriori informazioni, vedere [Privilegi di runtime e funzionalità Linux](#) sul sito Docker Docs.

Argomenti

- [Memorizzazione nella cache di Amazon S3](#)
- [Caching locale](#)
- [Specificare una cache locale](#)

Memorizzazione nella cache di Amazon S3

Il caching di Amazon S3 archivia la cache in un bucket Amazon S3 disponibile su più host di build. Questa è una buona opzione per artefatti di build di piccole e medie dimensioni che sono più costosi da creare che da scaricare.

Per utilizzare Amazon S3 in una build, puoi specificare i percorsi per i file che desideri memorizzare nella cache. `buildspec.yml` CodeBuild memorizzerà e aggiornerà automaticamente la cache nella posizione Amazon S3 configurata nel progetto. Se non specifichi i percorsi dei file, CodeBuild memorizzerà nella cache al meglio le dipendenze del linguaggio comune per aiutarti a velocizzare le build. Puoi visualizzare i dettagli della cache nei log di compilazione.

Inoltre, se desideri avere più versioni della cache, puoi definire una chiave di cache in `buildspec.yml`. CodeBuild memorizza la cache nel contesto di questa chiave di cache e crea una copia cache unica che non verrà aggiornata una volta creata. Le chiavi della cache possono essere

condivise anche tra progetti. Funzionalità come le chiavi dinamiche, il controllo delle versioni della cache e la condivisione della cache tra le build sono disponibili solo quando viene specificata una chiave.

Per ulteriori informazioni sulla sintassi della cache nel file `buildspec`, consulta il riferimento [buildspec.cache](#)

Argomenti

- [Genera chiavi dinamiche](#)
- [codebuild-hash-files](#)
- [Versione cache](#)
- [Condivisione della cache tra progetti](#)
- [Esempi di Buildspec](#)

Genera chiavi dinamiche

Una chiave cache può includere comandi di shell e variabili di ambiente per renderla unica, abilitando gli aggiornamenti automatici della cache quando la chiave cambia. Ad esempio, è possibile definire una chiave utilizzando l'hash del `package-lock.json` file. Quando le dipendenze in quel file cambiano, l'hash, e quindi la chiave della cache, cambia, attivando la creazione automatica di una nuova cache.

```
cache:
  key: npm-key-$(codebuild-hash-files package-lock.json)
```

CodeBuild valuterà l'espressione per ottenere la chiave finale `$(codebuild-hash-files package-lock.json)`:

```
npm-key-abc123
```

È inoltre possibile definire una chiave di cache utilizzando variabili di ambiente, ad esempio `CODEBUILD_RESOLVED_SOURCE_VERSION`. Ciò garantisce che ogni volta che la fonte cambia, venga generata una nuova chiave, con conseguente salvataggio automatico di una nuova cache:

```
cache:
```

```
key: npm-key-$CODEBUILD_RESOLVED_SOURCE_VERSION
```

CodeBuild valuterà l'espressione e otterrà la chiave finale:

```
npm-key-046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

codebuild-hash-files

`codebuild-hash-files` è uno strumento CLI che calcola un hash SHA-256 per un insieme di file nella directory di origine utilizzando modelli glob: CodeBuild

```
codebuild-hash-files <glob-pattern-1> <glob-pattern-2> ...
```

Ecco alcuni esempi che utilizzano: `codebuild-hash-files`

```
codebuild-hash-files package-lock.json
codebuild-hash-files '**/*.md'
```

Versione cache

La versione cache è un hash generato dai percorsi delle directory memorizzate nella cache. Se due cache hanno versioni diverse, vengono trattate come cache distinte durante il processo di abbinamento. Ad esempio, le due cache seguenti sono considerate diverse perché fanno riferimento a percorsi diversi:

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "pip-dependencies/**/*"
```

```
version: 0.2
```

```
phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target tmp/pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "tmp/pip-dependencies/**/*"
```

Condivisione della cache tra progetti

Puoi utilizzare il campo `cacheNamespace` API nella cache sezione per condividere una cache tra più progetti. Questo campo definisce l'ambito della cache. Per condividere una cache, devi fare quanto segue:

- Usa lo stesso `cacheNamespace`.
- Specificare la stessa `cachekey`.
- Definire percorsi di cache identici.
- Usa gli stessi bucket Amazon S3 e `pathPrefix` se impostati.

Ciò garantisce la coerenza e consente la condivisione della cache tra i progetti.

Specificare uno spazio dei nomi della cache (console)

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Seleziona Crea progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
3. In Artifacts, scegli Configurazione aggiuntiva.
4. Per il tipo di cache, scegli Amazon S3.
5. Per lo spazio dei nomi Cache, facoltativo, inserisci un valore per lo spazio dei nomi.

▼ Additional configuration

Cache, encryption key

Encryption key - optional

Provide the AWS KMS customer master key used to encrypt this build's output artifacts. The default is your AWS-managed customer master key for S3.

arn:aws:kms:<region-ID>:<account-ID>:key/<key-ID>

Cache type**Cache bucket****Cache path prefix - optional****Cache lifecycle (days) - optional**

You can apply a lifecycle expiration action to all or a subset of objects in the cache bucket based on the path prefix.

[+ Add expiration](#)**Cache namespace - optional**

Provide a cache namespace if you want to share caches across projects.

6. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.

Specificate uno spazio dei nomi della cache ()AWS CLI

È possibile utilizzare il `--cache` parametro in AWS CLI per specificare uno spazio dei nomi della cache.

```
--cache '{"type": "S3", "location": "your-s3-bucket", "cacheNamespace": "test-cache-namespace"}'
```

Esempi di Buildspec

Ecco alcuni esempi di buildspec per linguaggi comuni:

Argomenti

- [Memorizza le dipendenze di Node.js](#)
- [Dipendenze da Cache Python](#)
- [Cache delle dipendenze di Ruby](#)
- [Dipendenze Cache Go](#)

Memorizza le dipendenze di Node.js

Se il progetto include un `package-lock.json` file e lo utilizza npm per gestire le dipendenze di Node.js, l'esempio seguente mostra come impostare la memorizzazione nella cache. Per impostazione predefinita, npm installa le dipendenze nella directory `node_modules`

```
version: 0.2

phases:
  build:
    commands:
      - npm install
cache:
  key: npm-${codebuild-hash-files package-lock.json}
  paths:
    - "node_modules/**/*"
```

Dipendenze da Cache Python

Se il progetto include un `requirements.txt` file e utilizza pip per gestire le dipendenze di Python, l'esempio seguente dimostra come configurare la memorizzazione nella cache. Per impostazione predefinita, pip installa i pacchetti nella directory del sistema `site-packages`

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt
cache:
  key: python-${codebuild-hash-files requirements.txt}
  paths:
    - "/root/.pyenv/versions/${python_version}/lib/python${python_major_version}/site-packages/**/*"
```

Inoltre, puoi installare le dipendenze in una directory specifica e configurare la memorizzazione nella cache per quella directory.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt --target python-dependencies
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "python-dependencies/**/*"
```

Cache delle dipendenze di Ruby

Se il progetto include un `Gemfile.lock` file e lo utilizza Bundler per gestire le dipendenze gem, l'esempio seguente dimostra come configurare la memorizzazione nella cache in modo efficace.

```
version: 0.2

phases:
  build:
    commands:
      - bundle install --path vendor/bundle
cache:
  key: ruby-$(codebuild-hash-files Gemfile.lock)
  paths:
    - "vendor/bundle/**/*"
```

Dipendenze Cache Go

Se il progetto include un `go.sum` file e utilizza i moduli Go per gestire le dipendenze, l'esempio seguente dimostra come configurare la memorizzazione nella cache. Per impostazione predefinita, i moduli Go vengono scaricati e archiviati nella directory. `${GOPATH}/pkg/mod`

```
version: 0.2

phases:
  build:
    commands:
```

```
- go mod download
cache:
  key: go-$(codebuild-hash-files go.sum)
  paths:
    - "/go/pkg/mod/**/*"
```

Caching locale

Questa modalità archivia la cache in locale su un host di compilazioni disponibile solo per tale host. Questa è una buona opzione per gli artefatti di build di dimensioni intermedie e grandi perché la cache è immediatamente disponibile sull'host di build. Questa non è l'opzione migliore se le compilazioni avvengono raramente. Ciò significa che il trasferimento su rete non influisce sulle prestazioni delle compilazioni.

Se decidi di optare per il caching locale, devi scegliere una o più delle seguenti modalità cache:

- La modalità cache delle origini memorizza i metadati Git per origini primarie e secondarie. Dopo la creazione della cache, le compilazioni successive estraggono solo le modifiche tra i commit. Questa modalità rappresenta una scelta valida per progetti con una directory di lavoro pulita e un'origine costituita da un repository Git di grandi dimensioni. Se scegli questa opzione e il tuo progetto non utilizza un repository Git (AWS CodeCommit, GitHub, GitHub Enterprise Server o Bitbucket), l'opzione viene ignorata.
- La modalità cache di livello Docker memorizza i livelli Docker esistenti. Rappresenta la soluzione ideale per progetti che prevedono la creazione o l'estrazione di immagini Docker di grandi dimensioni. Può inoltre prevenire eventuali problemi di prestazioni causati dall'estrazione di immagini Docker di grandi dimensioni dalla rete.

Note

- La cache di livello Docker può essere utilizzata solo in ambiente Linux.
- È necessario impostare il flag `privileged` in modo che il progetto disponga delle autorizzazioni Docker richieste.

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

- Prima di utilizzare una cache di livello Docker, è consigliabile considerare le implicazioni per la sicurezza.
-
- La modalità cache personalizzata memorizza le directory specificate nel file di specifiche di compilazione. Rappresenta una buona scelta se lo scenario di compilazione non è idoneo per nessuna delle altre due modalità di cache locale. Se utilizzi una cache personalizzata:
 - Puoi specificare per il caching solo le directory. Non puoi specificare singoli file.
 - Per fare riferimento alle directory memorizzate nella cache si utilizzano i symlink.
 - Le directory memorizzate nella cache vengono collegate alla compilazione prima del download delle origini del progetto. Gli elementi memorizzati nella cache sostituiscono gli elementi di origine se hanno lo stesso nome. Le directory vengono specificate utilizzando i percorsi di cache nel file di specifiche di compilazione. Per ulteriori informazioni, consulta [Sintassi buildspec](#).
 - Evitare nomi di directory uguali nell'origine e nella cache. Le directory memorizzate nella cache locale possono sovrascrivere o eliminare il contenuto delle directory nel repository di origine con lo stesso nome.

Note

La memorizzazione nella cache locale non è supportata con il tipo di LINUX_GPU_CONTAINER ambiente e il tipo di BUILD_GENERAL1_2XLARGE elaborazione. Per ulteriori informazioni, consulta [Modi e tipi di calcolo dell'ambiente di creazione](#).

Note

La memorizzazione nella cache locale non è supportata quando si configura CodeBuild per lavorare con un VPC. Per ulteriori informazioni sull'utilizzo VPCs con CodeBuild, consulta [Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud](#)

Specificare una cache locale

È possibile utilizzare la console AWS CLI, l'SDK o CloudFormation specificare una cache locale. Per ulteriori informazioni sulla memorizzazione nella cache locale, consulta [Caching locale](#)

Argomenti

- [Specifica del caching locale \(CLI\)](#)
- [Specifica del caching locale \(console\)](#)
- [Specifica del caching locale \(CloudFormation\)](#)

Specifica del caching locale (CLI)

È possibile utilizzare il `--cache` parametro in AWS CLI per specificare ciascuno dei tre tipi di cache locale.

- Per specificare una cache delle origini:

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Per specificare una cache di livello Docker:

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

- Per specificare una cache personalizzata:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).

Specifica del caching locale (console)

Puoi specificare una cache nella sezione Artifacts (Artefatti) della console. Per il tipo di cache, scegli Amazon S3 o Local. Se selezioni Local (Locale), scegli una o più delle tre opzioni di cache locale.

Cache type

Local ▼

Select one or more local cache options.

- ☐ **Docker layer cache**
Caches existing Docker layers so they can be reused. Requires privileged mode.
- ☐ **Source cache**
Caches .git metadata so subsequent builds only pull the change in commits.
- ☐ **Custom cache**
Caches directories specified in the buildspec file.

Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

Specifica del caching locale (CloudFormation)

Se si utilizza CloudFormation per specificare una cache locale, nella Cache proprietà, perType, specificareLOCAL. Il seguente CloudFormation codice in formato YAML di esempio specifica tutti e tre i tipi di cache locale. È possibile selezionare i tipi disponibili in qualsiasi combinazione. Se utilizzi una cache di livello Docker, in Environment devi impostare PrivilegedMode su true e Type su LINUX_CONTAINER.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: <service-role>
    Artifacts:
      Type: S3
      Location: <bucket-name>
      Name: myArtifact
      EncryptionDisabled: true
      OverrideArtifactName: true
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
      Certificate: <bucket/cert.zip>
      # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
      PrivilegedMode: true
    Source:
```

```
Type: GITHUB
Location: <github-location>
InsecureSsl: true
GitCloneDepth: 1
ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
  Type: LOCAL
  Modes: # You can specify one or more cache mode,
    - LOCAL_CUSTOM_CACHE
    - LOCAL_DOCKER_LAYER_CACHE
    - LOCAL_SOURCE_CACHE
```

Note

Per impostazione predefinita, il daemon Docker è abilitato per le build non VPC. Se desideri utilizzare i contenitori Docker per le build VPC, [consulta Runtime Privilege e Linux Capabilities sul sito Web di Docker Docs](#) e abilita la modalità privilegiata. Inoltre, Windows non supporta la modalità privilegiata.

Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(CloudFormation\)](#).

Il debug si integra AWS CodeBuild

AWS CodeBuild fornisce due metodi per il debug delle build durante lo sviluppo e la risoluzione dei problemi. È possibile utilizzare l'ambiente CodeBuild Sandbox per esaminare i problemi e convalidare le correzioni in tempo reale oppure utilizzare AWS Systems Manager Session Manager per connettersi al contenitore di build e visualizzare lo stato del contenitore.

Esegui il debug delle build con sandbox CodeBuild

L'ambiente CodeBuild sandbox fornisce una sessione di debug interattiva in un ambiente sicuro e isolato. È possibile interagire con l'ambiente direttamente tramite la sala operatoria Console di gestione AWS AWS CLI, eseguire comandi e convalidare il processo di compilazione passo dopo passo. Utilizza un modello di fatturazione al secondo conveniente e supporta la stessa integrazione nativa con i fornitori e AWS i servizi di origine dell'ambiente di compilazione. Puoi anche connetterti a un ambiente sandbox utilizzando client SSH o dai tuoi ambienti di sviluppo integrati (). IDEs

[Per ulteriori informazioni sui prezzi della CodeBuild sandbox, consulta la CodeBuild documentazione sui prezzi.](#) Per istruzioni dettagliate, consulta la [Compilazioni di debug con sandbox CodeBuild](#) documentazione.

Compilazioni di debug con Session Manager

AWS Systems Manager Session Manager consente l'accesso diretto alle build in esecuzione nel loro ambiente di esecuzione effettivo. Questo approccio consente di connettersi a contenitori di build attivi e di ispezionare il processo di compilazione in tempo reale. È possibile esaminare il file system, monitorare i processi in esecuzione e risolvere i problemi non appena si verificano.

Per istruzioni dettagliate, consulta la [Compilazioni di debug con Session Manager](#) documentazione.

Compilazioni di debug con sandbox CodeBuild

In AWS CodeBuild, puoi eseguire il debug di una build utilizzando CodeBuild sandbox per eseguire comandi personalizzati e risolvere i problemi della build.

Argomenti

- [Prerequisiti](#)
- [Esegui il debug delle build con CodeBuild sandbox \(console\)](#)
- [Compila il debug con CodeBuild sandbox \(\)AWS CLI](#)
- [Tutorial: Connessione a una sandbox tramite SSH](#)
- [Risoluzione dei problemi AWS CodeBuild di connessione SSH nella sandbox](#)

Prerequisiti

Prima di utilizzare una CodeBuild sandbox, assicurati che il tuo ruolo di CodeBuild servizio abbia la seguente politica SSM:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
        "Effect": "Allow",
        "Action": [
            "ssmmessages:CreateControlChannel",
            "ssmmessages:CreateDataChannel",
            "ssmmessages:OpenControlChannel",
            "ssmmessages:OpenDataChannel"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ssm:StartSession"
        ],
        "Resource": [
            "arn:aws:codebuild:us-east-1:111122223333:build/*",
            "arn:aws:ssm:us-east-1::document/AWS-StartSSHSession"
        ]
    }
]
```

Esegui il debug delle build con CodeBuild sandbox (console)

Usa le seguenti istruzioni per eseguire comandi e connettere il tuo client SSH alla CodeBuild sandbox nella console.

Esegui i comandi con CodeBuild sandbox (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Scegli il progetto di compilazione, quindi scegli Debug build.

sandbox-project Actions Create trigger Edit Clone Clear cache Debug build Start build with overrides Start build

Configuration

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::...:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

Build history Batch history **Project details** Build triggers Metrics Debug sessions

Project configuration

 Edit

Name	Description
sandbox-project	-
Project ARN	Build badge
☐ arn:aws:codebuild:us-east-1::project/sandbox-project	Disabled

3. Nella scheda Esegui comando, inserisci i comandi personalizzati, quindi scegli Esegui comando.

Debug build

Run Command SSH Client Session Manager

Run custom commands with sandbox Learn more

- Launches a sandbox environment mirroring your project configuration.
- Automatically downloads source code, while skipping project buildspec execution.
- Ideal for reproducing failure, experimenting fixes and investigation.

Command

```
1 pwd
```

1:4 SH

Run command

4. La CodeBuild sandbox verrà quindi inizializzata e inizierà a eseguire i comandi personalizzati. L'output verrà visualizzato nella scheda Output una volta completato.

Debug build

[Run Command](#)[SSH Client](#)[Session Manager](#)**Sandbox is running**Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is ready and available for use.[Stop sandbox](#)

Command

1 `pwd`

⊞ 0 ⚠ 0

1:1 SH

[Run command](#)[Command output](#)[Sandbox phases](#)[Sandbox logs](#)[Sandbox configurations](#)[Command history](#)[View entire log in CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

5. Una volta completata la risoluzione dei problemi, puoi interrompere la sandbox scegliendo **Stop sandbox**. Quindi scegli **Stop** per confermare che la sandbox verrà interrotta.

Stop sandbox



Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

- `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18`

[Cancel](#)[Stop](#)

Debug build

[Run Command](#)[SSH Client](#)[Session Manager](#)**Sandbox is stopped**

Your sandbox `sandbox-project:ef8f3204-a9e8-4707-afcf-b4bb49b6bc18` is currently inactive.

[Command output](#)[Sandbox phases](#)[Sandbox logs](#)[Sandbox configurations](#)[Command history](#)[View entire log in CloudWatch console](#)

```
1 /codebuild/output/src3141870147/src
2
```

Connect al client SSH con CodeBuild sandbox (console)

1. [Apri la AWS CodeBuild console su codebuild/homehttps://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/).
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Scegli il progetto di compilazione, quindi scegli Debug build.

sandbox-project

Actions ▾Create triggerEditCloneClear cacheDebug buildStart build with overridesStart build

Configuration

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::[redacted]:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

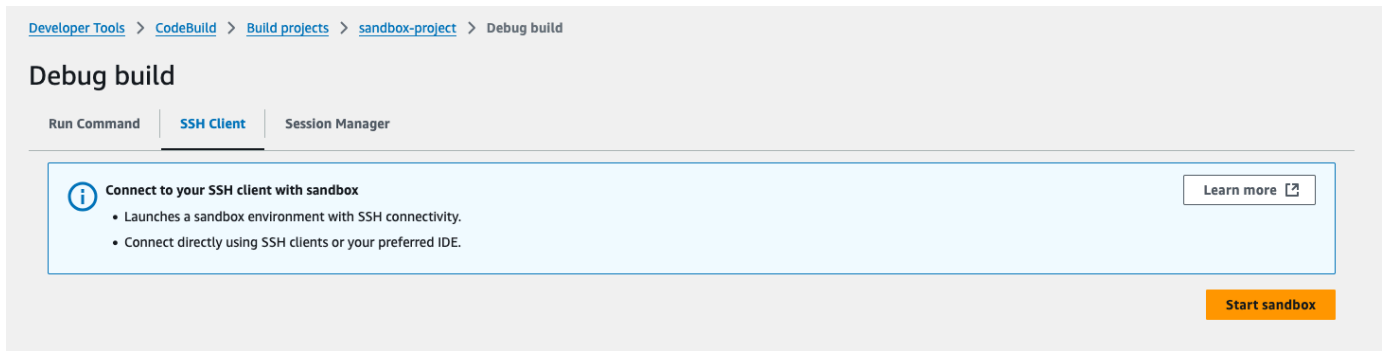
Build historyBatch historyProject detailsBuild triggersMetricsDebug sessions

Project configuration

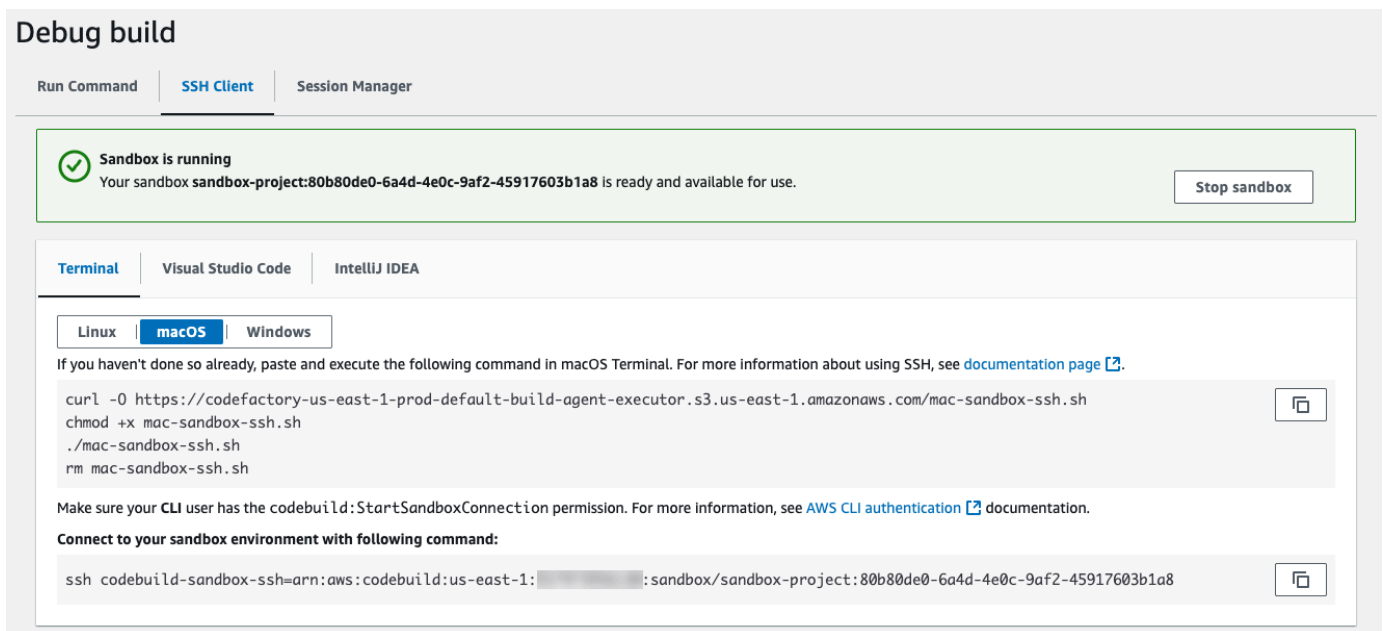
Edit

Name	Description
sandbox-project	-
Project ARN	Build badge
☐ arn:aws:codebuild:us-east-1:[redacted]:project/sandbox-project	Disabled

3. Nella scheda Client SSH, scegli Avvia sandbox.



4. Dopo l'avvio della CodeBuild sandbox, segui le istruzioni della console per connettere il client SSH alla sandbox.



5. Una volta completata la risoluzione dei problemi, puoi interrompere la sandbox scegliendo Stop sandbox. Quindi scegli Stop per confermare che la sandbox verrà interrotta.

Stop sandbox

×

Stopping this sandbox will terminate all active sessions and running commands. Are you sure you want to stop the sandbox?

• sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8

Cancel

Stop

Debug build

Run CommandSSH ClientSession Manager

⚠

Sandbox is stopped

Your sandbox `sandbox-project:80b80de0-6a4d-4e0c-9af2-45917603b1a8` is currently inactive.

Sandbox phasesSandbox logsSandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
QUEUED	✔ Succeeded	-	<1 sec	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
PROVISIONING	✔ Succeeded		5 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	✔ Succeeded		8 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:33 PM (UTC-7:00)
RUN_SANDBOX	✔ Succeeded		213 secs	Apr 8, 2025 1:33 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
UPLOAD_ARTIFACTS	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
FINALIZING	✔ Succeeded		<1 sec	Apr 8, 2025 1:36 PM (UTC-7:00)	Apr 8, 2025 1:36 PM (UTC-7:00)
COMPLETED	✔ Succeeded	-	-	Apr 8, 2025 1:36 PM (UTC-7:00)	-

Compila il debug con CodeBuild sandbox ()AWS CLI

Usa le seguenti istruzioni per eseguire comandi e connettere il tuo client SSH alla sandbox.
CodeBuild

Avvia una CodeBuild sandbox ()AWS CLI

CLI command

```
aws codebuild start-sandbox --project-name $PROJECT_NAME
```

- `--project-name`: nome CodeBuild del progetto

Sample request

```
aws codebuild start-sandbox --project-name "project-name"
```

Sample response

```
{
  "id": "project-name",
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
  "projectName": "project-name",
  "requestTime": "2025-02-06T11:24:15.560000-08:00",
  "status": "QUEUED",
  "source": {
    "type": "S3",
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-sources/eb-sample-jetty-v4.zip",
    "insecureSsl": false
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:6.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [{
      "name": "foo",
      "value": "bar",
      "type": "PLAINTEXT"
    },
    {
      "name": "bar",
      "value": "baz",
      "type": "PLAINTEXT"
    }
  ],
  "privilegedMode": false,
  "imagePullCredentialsType": "CODEBUILD"
},
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
```

```

        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "QUEUED",
    "status": "QUEUED",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "logs": {
        "groupName": "group",
        "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
            "encryptionDisabled": false
        }
    }
}
}

```

```
}
```

Ottieni informazioni sullo stato della sandbox ()AWS CLI

CLI command

```
aws codebuild batch-get-sandboxes --ids $SANDBOX_IDS
```

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

- `--ids`: elenco separato da virgole di `sandboxIds` o `sandboxArns`.

Puoi fornire un ID sandbox o un ARN sandbox:

- ID sandbox: *<codebuild-project-name>: <UUID>*

Ad esempio, `project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

- ARN della sandbox: `arn:aws:codebuild: <region>: <account-id> <codebuild-project-name> <UUID>`

Ad esempio, `arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

Sample response

```
{  
  "sandboxes": [{  
    "id": "project-name",  
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
    "projectName": "project-name",  
    "requestTime": "2025-02-06T11:24:15.560000-08:00",  
    "endTime": "2025-02-06T11:39:21.587000-08:00",  
    "status": "STOPPED",  
    "source": {  
      "type": "S3",
```

```

        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
            "name": "foo",
            "value": "bar",
            "type": "PLAINTEXT"
        },
        {
            "name": "bar",
            "value": "baz",
            "type": "PLAINTEXT"
        }
    ],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"timeoutInMinutes": 10,
"queuedTimeoutInMinutes": 480,
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
},
"encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/
SampleEncryptionKey",
"serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
"currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "COMPLETED",
    "status": "STOPPED",

```

```
"startTime": "2025-02-06T11:24:15.626000-08:00",
"endTime": "2025-02-06T11:39:21.600000-08:00",
"phases": [{
  "phaseType": "SUBMITTED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.577000-08:00",
  "endTime": "2025-02-06T11:24:15.606000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "QUEUED",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:15.606000-08:00",
  "endTime": "2025-02-06T11:24:16.067000-08:00",
  "durationInSeconds": 0
},
{
  "phaseType": "PROVISIONING",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:16.067000-08:00",
  "endTime": "2025-02-06T11:24:20.519000-08:00",
  "durationInSeconds": 4,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "DOWNLOAD_SOURCE",
  "phaseStatus": "SUCCEEDED",
  "startTime": "2025-02-06T11:24:20.519000-08:00",
  "endTime": "2025-02-06T11:24:22.238000-08:00",
  "durationInSeconds": 1,
  "contexts": [{
    "statusCode": "",
    "message": ""
  }]
},
{
  "phaseType": "RUNNING_SANDBOX",
  "phaseStatus": "TIMED_OUT",
  "startTime": "2025-02-06T11:24:22.238000-08:00",
  "endTime": "2025-02-06T11:39:21.560000-08:00",
  "durationInSeconds": 899,
```



```

        "contexts": [{
            "statusCode": "BUILD_TIMED_OUT",
            "message": "Build has timed out. "
        }]
    },
    {
        "phaseType": "COMPLETED",
        "startTime": "2025-02-06T11:39:21.560000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}
}],
"sandboxesNotFound": []
}

```

Ferma una sandbox (AWS CLI)

CLI command

```
aws codebuild stop-sandbox --id $SANDBOX-ID
```

- `--id`: A sandboxId o. sandboxArn

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/  
project-name"
```

Sample response

```
{  
  "id": "project-name",  
  "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",  
  "projectName": "project-name",  
  "requestTime": "2025-02-06T11:24:15.560000-08:00",  
  "status": "STOPPING",  
  "source": {  
    "type": "S3",  
    "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-  
sources/eb-sample-jetty-v4.zip",  
    "insecureSsl": false  
  },  
  "environment": {  
    "type": "LINUX_CONTAINER",  
    "image": "aws/codebuild/standard:6.0",  
    "computeType": "BUILD_GENERAL1_SMALL",  
    "environmentVariables": [{  
      "name": "foo",  
      "value": "bar",  
      "type": "PLAINTEXT"  
    },  
    {  
      "name": "bar",  
      "value": "baz",  
      "type": "PLAINTEXT"  
    }  
  ],  
}
```

```

    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
  },
  "timeoutInMinutes": 10,
  "queuedTimeoutInMinutes": 480,
  "logConfig": {
    "cloudWatchLogs": {
      "status": "ENABLED",
      "groupName": "group",
      "streamName": "stream"
    },
    "s3Logs": {
      "status": "ENABLED",
      "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
      "encryptionDisabled": false
    }
  },
  "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
  "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
  "currentSession": {
    "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "currentPhase": "RUN_SANDBOX",
    "status": "STOPPING",
    "startTime": "2025-02-06T11:24:15.626000-08:00",
    "phases": [{
      "phaseType": "SUBMITTED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.144000-08:00",
      "endTime": "2025-02-08T14:33:26.173000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "QUEUED",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.173000-08:00",
      "endTime": "2025-02-08T14:33:26.702000-08:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "PROVISIONING",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2025-02-08T14:33:26.702000-08:00",
      "endTime": "2025-02-08T14:33:30.530000-08:00",
      "durationInSeconds": 3,

```

```

        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "DOWNLOAD_SOURCE",
        "phaseStatus": "SUCCEEDED",
        "startTime": "2025-02-08T14:33:30.530000-08:00",
        "endTime": "2025-02-08T14:33:33.478000-08:00",
        "durationInSeconds": 2,
        "contexts": [{
            "statusCode": "",
            "message": ""
        }]
    },
    {
        "phaseType": "RUN_SANDBOX",
        "startTime": "2025-02-08T14:33:33.478000-08:00"
    }
],
"logs": {
    "groupName": "group",
    "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
        "encryptionDisabled": false
    }
}

```

```
    }  
  }  
}
```

Avvia l'esecuzione di un comando (AWS CLI)

CLI command

```
aws codebuild start-command-execution --command $COMMAND --type $TYPE --sandbox-id  
$SANDBOX-ID
```

- `--command`: Il comando che deve essere eseguito.
- `--sandbox-id`: A sandboxId osandboxArn.
- `--type`: Il tipo di comando, SHELL.

Sample request

```
aws codebuild start-command-execution --command "echo \"Hello World\"" --type SHELL --  
sandbox-id "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{  
  "id": "e1c658c2-02bb-42a8-9abb-94835241fcd6",  
  "sandboxId": "f7126a4a-b0d5-452f-814c-fea73718f805",  
  "submitTime": "2025-02-06T20:12:02.683000-08:00",  
  "status": "SUBMITTED",  
  "command": "echo \"Hello World\"",  
  "type": "SHELL",  
  "logs": {  
    "groupName": "group",  
    "streamName": "stream",  
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-  
west-2#logsV2:log-groups/log-group/group/log-events/stream",  
    "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-  
pool-1-us-west-2-beta-default-build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz?  
region=us-west-2",  
    "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-  
group:group:log-stream:stream",  
  }  
}
```

```

    "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-
build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz",
    "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
    },
    "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
    }
}
}

```

Ottieni informazioni sulle esecuzioni dei comandi ()AWS CLI

CLI command

```
aws codebuild batch-get-command-executions --command-execution-ids $COMMAND-IDS --
sandbox-id $SANDBOX-IDS
```

- `--command-execution-ids`: elenco separato da virgole di `commandExecutionIds`
- `--sandbox-id`: A `sandboxId` o `sandboxArn`.

Sample request

```
aws codebuild batch-get-command-executions --command-execution-
ids "c3c085ed-5a8f-4531-8e95-87d547f27ffd" --sandbox-id "arn:aws:codebuild:us-
west-2:962803963624:sandbox/project-name"
```

Sample response

```

{
  "commandExecutions": [{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",

```

```

    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
        "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
        "encryptionDisabled": false
      }
    },
    "commandExecutionsNotFound": []
  }
}

```

Elenca le esecuzioni di comandi per una sandbox ()AWS CLI

CLI command

```
aws codebuild list-command-executions-for-sandbox --sandbox-id $SANDBOX-ID --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --sort-order $SORT_ORDER
```

- `--next-token`: Il token successivo, se presente, per ottenere risultati impaginati. Otterrete questo valore dalla precedente esecuzione delle sandbox di elenchi.

- `--max-results`: (Facoltativo) Il numero massimo di record sandbox da recuperare.
- `--sort-order`: L'ordine in cui i record della sandbox devono essere recuperati.

Sample request

```
aws codebuild list-command-executions-for-sandbox --sandbox-id
"arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
  "commandExecutions": [{
    "id": "aad6687e-07bc-45ab-a1fd-f5440229b528",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:35.304000-08:00",
    "startTime": "2025-02-10T20:18:35.615000-08:00",
    "endTime": "2025-02-10T20:18:35.651000-08:00",
    "status": "FAILED",
    "command": "fail command",
    "type": "SHELL",
    "exitCode": "127",
    "standardErrContent": "/codebuild/output/tmp/script.sh: 4: fail: not
found\n",
    "logs": {
      "groupName": "group",
      "streamName": "stream",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
      "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
      "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
      "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
      "cloudWatchLogs": {
        "status": "ENABLED",
        "groupName": "group",
        "streamName": "stream"
      },
      "s3Logs": {
        "status": "ENABLED",
```



```

        "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
        "encryptionDisabled": false
    }
},
{
    "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
    "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
    "submitTime": "2025-02-10T20:18:17.118000-08:00",
    "startTime": "2025-02-10T20:18:17.939000-08:00",
    "endTime": "2025-02-10T20:18:17.976000-08:00",
    "status": "SUCCEEDED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "exitCode": "0",
    "standardOutputContent": "Hello World\n",
    "logs": {
        "groupName": "group",
        "streamName": "stream",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
            "encryptionDisabled": false
        }
    }
}
]

```

```
}
```

Elenca le sandbox (AWS CLI)

CLI command

```
aws codebuild list-sandboxes --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --  
sort-order $SORT_ORDER
```

Sample request

```
aws codebuild list-sandboxes
```

Sample response

```
{  
  "ids": [  
    "s3-log-project-integ-test-temp173925062814985d64e0f-7880-41df-9a3c-  
fb6597a266d2:827a5243-0841-4b69-a720-4438796f6967",  
    "s3-log-project-integ-test-temp1739249999716bbd438dd-8bb8-47bd-  
ba6b-0133ac65b3d3:e2fa4eab-73af-42e3-8903-92fddaf9f378",  
    "s3-log-project-integ-test-  
temp17392474779450fbdacc2-2d6e-4190-9ad5-28f891bb7415:cd71e456-2a4c-4db4-ada5-  
da892b0bba05",  
    "s3-log-project-integ-test-temp17392246284164301421c-5030-4fa1-b4d3-  
ca15e44771c5:9e26ab3f-65e4-4896-a19c-56b1a95e630a",  
    "s3-log-project-integ-test-temp173921367319497056d8d-6d8e-4f5a-a37c-  
a62f5686731f:22d91b06-df1e-4e9c-a664-c0abb8d5920b",  
    "s3-log-project-integ-test-temp1739213439503f6283f19-390c-4dc8-95a9-  
c8480113384a:82cc413e-fc46-47ab-898f-ae23c83a613f",  
    "s3-log-project-integ-test-temp1739054385570b1f1ddc2-0a23-4062-  
bd0c-24e9e4a99b99:c02562f3-2396-42ec-98da-38e3fe5da13a",  
    "s3-log-project-integ-test-temp173905400540237dab1ac-1fde-4dfb-a8f5-  
c0114333dc89:d2f30493-f65e-4fa0-a7b6-08a5e77497b9",  
    "s3-log-project-integ-test-  
temp17390534055719c534090-7bc4-48f1-92c5-34acaec5bf1e:df5f1c8a-f017-43b7-91ba-  
ad2619e2c059",  
    "s3-log-project-integ-test-temp1739052719086a61813cc-  
ebb9-4db4-9391-7f43cc984ee4:d61917ec-8037-4647-8d52-060349272c4a",  
    "s3-log-project-integ-test-temp173898670094078b67edb-  
c42f-42ed-9db2-4b5c1a5fc66a:ce33dfbc-beeb-4466-8c99-a3734a0392c7",  
  ]  
}
```

```
"s3-log-project-integ-test-  
temp17389863425584d21b7cd-32e2-4f11-9175-72c89ecaffef:046dadf0-1f3a-4d51-a2c0-  
e88361924acf",  
  "s3-log-project-integ-test-  
temp1738985884273977ccd23-394b-46cc-90d3-7ab94cf764dc:0370dc41-9339-4b0a-91ed-51929761b244",  
  "s3-log-project-integ-test-temp1738985365972241b614f-8e41-4387-  
bd25-2b8351fbc9e0:076c392a-9630-47d8-85a9-116aa34edfff",  
  "s3-log-project-integ-test-  
temp1738985043988a51a9e2b-09d6-4d24-9c3c-1e6e21ac9fa8:6ea3949c-435b-4177-  
aa4d-614d5956244c",  
  "s3-log-project-integ-test-temp1738984123354c68b31ad-49d1-4f4b-981d-  
b66c00565ff6:6c3fff6c-815b-48b5-ada3-737400a6dee8",  
  "s3-log-project-integ-test-  
temp1738977263715d4d5bf6c-370a-48bf-8ea6-905358a6cf92:968a0f54-724a-42d1-9207-6ed854b2fae8",  
  "s3-log-project-integ-test-  
temp173897358796816ce8d7d-2a5e-41ef-855b-4a94a8d2795d:80f9a7ce-930a-402e-934e-  
d8b511d68b04",  
  "s3-log-project-integ-test-temp17389730633301af5e452-0966-467c-  
b684-4e36d47f568c:cabbe989-2e8a-473c-af25-32edc8c28646",  
  "s3-log-project-integ-test-temp1738901503813173fd468-  
b723-4d7b-9f9f-82e88d17f264:f7126a4a-b0d5-452f-814c-fea73718f805",  
  "s3-log-project-integ-test-temp1738890502472c13616fb-  
bd0f-4253-86cc-28b74c97a0ba:c6f197e5-3a53-45b6-863e-0e6353375437",  
  "s3-log-project-integ-test-  
temp17388903044683610daf3-8da7-43c6-8580-9978432432ce:d20aa317-8838-4966-  
bbfc-85b908213df1",  
  "s3-log-project-integ-test-temp173888857196780b5ab8b-e54b-44fd-a222-  
c5a374fffe96:ab4b9970-ffae-47a0-b3a8-7b6790008cad",  
  "s3-log-project-integ-test-temp1738888336931c11d378d-e74d-49a4-  
a723-3b92e6f7daac:4922f0e8-9b7d-4119-9c9f-115cd85e703e",  
  "s3-log-project-integ-test-temp17388881717651612a397-c23f-4d88-  
ba87-2773cd3fc0c9:be91c3fc-418e-4feb-8a3a-ba58ff8f4e8a",  
  "s3-log-project-integ-test-  
temp17388879727174c3c62ed-6195-4afb-8a03-59674d0e1187:a48826a8-3c0d-43c5-  
a1b5-1c98a0f978e9",  
  "s3-log-project-integ-test-temp1738885948597cef305e4-b8b4-46b0-a65b-  
e2d0a7b83294:c050e77d-e3f8-4829-9a60-46149628fe96",  
  "s3-log-project-integ-test-temp173888561463001a7d2a8-  
e4e4-4434-94db-09d3da9a9e17:8c3ac3f5-7111-4297-aec9-2470d3ead873",  
  "s3-log-project-integ-test-  
temp1738869855076eb19cafd-04fe-41bd-8aa0-40826d0c0d27:d25be134-05cb-404a-85da-  
ac5f85d2d72c",  
  "s3-project-integ-test-temp1738868157467148eacfc-d39b-49fc-a137-  
e55381cd2978:4909557b-c221-4814-b4b6-7d9e93d37c35",
```

```

        "s3-project-integ-test-temp1738820926895abec0af2-
e33d-473c-9cf4-2122dd9d6876:8f5cf218-71d6-40a4-a4be-6cacebd7765f",
        "s3-project-integ-test-temp173881998877574f969a6-1c2e-4441-b463-
ab175b45ce32:04396851-c901-4986-9117-585528e3877f",
        "s3-project-integ-test-temp17388189812309abd2604-29ba-4cf6-
b6bf-073207b7db9c:540075c7-f5ec-41e8-9341-2233c09247eb",
        "s3-project-integ-test-temp1738818843474d3ea9ac1-b609-461b-
bbdb-2da245c9bc96:865d4c3c-fbfe-4ece-9c92-d0c928341404",
        "s3-project-integ-test-temp1738818542236006e9169-e6d9-4344-9b59-
f557e7aec619:1f9ffa87-da15-4290-83e2-eebdd877497b",
        "s3-project-integ-test-
temp173881809557486ad11fd-7931-48d7-81d5-499cea52a6bc:c4c2efc4-685f-4e13-8b0f-1ef85ec300b1",
        "s3-project-integ-test-temp173881794103322941020-3f0b-49c3-b836-
fcd818ec9484:0344cfba-de48-456d-b2a8-6566bd4a5d6e",
        "s3-project-integ-test-temp1738817680747b93d0d0b-ea16-497f-9559-
af25ee6dcfdf:654a3a55-d92a-4dc6-8da8-56fd4d40d7e1",
        "s3-project-integ-test-temp17388174027191255c3da-086c-4270-b047-
acac0b7bee0d:b7e82740-2c69-42fc-ab5a-dbf15bc016a1",
        "s3-project-integ-test-temp1738817099799016e7fa3-b9b5-46a2-
bcd5-0888c646743f:8705a6a4-79ff-427a-a1c3-85c4e8fe462e",
        "s3-project-integ-test-temp1738816479281bb0c3606-5ebf-4623-
bed5-12b60e9d3512:f23fc74b-a981-4835-8e28-375fcd4c99e4",
        "s3-project-integ-test-
temp1738816263585c939a133-4d37-482c-9238-1dbff34b7674:ca28e234-0045-4ae6-8732-938b17597f50",
        "s3-project-integ-test-
temp173881580873072d18733-8fe4-43b1-83f7-95f25bb27ccf:c6f0f55b-5736-47c7-
a3aa-1b8461a6d5ed"
    ]
}

```

Tutorial: Connessione a una sandbox tramite SSH

Questo tutorial mostra come connettersi a una CodeBuild sandbox utilizzando un client SSH.

Per completare questo tutorial, è necessario innanzitutto:

- Assicurati di avere un progetto esistente AWS CodeBuild .
- Imposta le autorizzazioni IAM appropriate configurate per il ruolo CodeBuild del tuo progetto.
- Installa e configura AWS CLI sul tuo computer locale.

Fase 1: Avvia una sandbox

Per avviare una CodeBuild sandbox nella console

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Scegli il progetto di compilazione, quindi scegli Debug build.

The screenshot shows the AWS CodeBuild console interface for a project named 'sandbox-project'. The breadcrumb navigation at the top reads: [Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > [sandbox-project](#). Below the project name, there are several action buttons: 'Actions' (with a dropdown arrow), 'Create trigger', 'Edit', 'Clone', 'Debug build', 'Start build with overrides', and a prominent orange 'Start build' button. The 'Configuration' section contains a table with the following details:

Source provider	Primary repository	Artifacts upload location	Service role
No source	-	-	arn:aws:iam::012345678910:role/service-role/codebuild-sandbox-project-service-role
Public builds	Disabled		

Below the configuration table, there are tabs for 'Build history', 'Batch history', 'Project details' (which is selected), 'Build triggers', 'Metrics', and 'Debug sessions'. The 'Project configuration' section shows the project's name as 'sandbox-project' and its ARN as 'arn:aws:codebuild:us-east-1:012345678910:project/sandbox-project'. The 'Build badge' is currently disabled. An 'Edit' button is located in the top right corner of this section.

3. Nella scheda Client SSH, scegli Avvia sandbox.

The screenshot shows the 'Debug build' page in the AWS CodeBuild console. The breadcrumb navigation is: [Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > [sandbox-project](#) > [Debug build](#). The page has three tabs: 'Run Command', 'SSH Client' (which is selected), and 'Session Manager'. A light blue information box contains the following text:

Connect to your SSH client with sandbox

- Launches a sandbox environment with SSH connectivity.
- Connect directly using SSH clients or your preferred IDE.

A 'Learn more' link with an external icon is located to the right of the list. At the bottom right of the page, there is an orange 'Start sandbox' button.

4. Il processo di inizializzazione della sandbox potrebbe richiedere del tempo. Puoi connetterti alla sandbox quando il suo stato cambia in. RUN_SANDBOX

Developer Tools > CodeBuild > Build projects > sandbox-project > Debug build

Debug build

Run CommandSSH ClientSession Manager

✓ Sandbox is running

Your sandbox sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256 is ready and available for use.

Stop sandbox

TerminalVisual Studio CodeIntelliJ IDEA

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Make sure your CLI user has the codebuild:StartSandboxConnection permission. For more information, see [AWS CLI authentication](#) documentation.

Connect to your sandbox environment with following command:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:012345678910:sandbox/sandbox-project:253616fd-9624-434e-bb9a-bbe52620d256
```

Sandbox phasesSandbox logsSandbox configurations

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✓ Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
QUEUED	✓ Succeeded	-	<1 sec	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
PROVISIONING	✓ Succeeded		4 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
DOWNLOAD_SOURCE	✓ Succeeded		6 secs	Apr 1, 2025 4:33 PM (UTC-7:00)	Apr 1, 2025 4:33 PM (UTC-7:00)
RUN_SANDBOX	-	-	-	Apr 1, 2025 4:33 PM (UTC-7:00)	-

Fase 2: Modificare la configurazione SSH locale

Se ti connetti alla sandbox per la prima volta, devi eseguire una procedura di configurazione una tantum utilizzando i seguenti passaggi:

Per modificare la configurazione SSH locale nella console

1. Individua i comandi di configurazione per il tuo sistema operativo.

2. Apri il terminale locale, quindi copia ed esegui i comandi forniti per scaricare ed eseguire lo script per configurare la configurazione SSH locale. Ad esempio, se il tuo sistema operativo è macOS, usa il seguente comando:

LinuxmacOSWindows

If you haven't done so already, paste and execute the following command in macOS Terminal. For more information about using SSH, see [documentation page](#).

```
curl -O https://codefactory-us-east-1-prod-default-build-agent-executor.s3.us-east-1.amazonaws.com/mac-sandbox-ssh.sh
chmod +x mac-sandbox-ssh.sh
./mac-sandbox-ssh.sh
rm mac-sandbox-ssh.sh
```

Compilazioni di debug con sandbox CodeBuild

Versione API 2016-10-06 691

3. Lo script di configurazione aggiungerà le configurazioni richieste per la connessione alle sandbox. Ti verrà richiesto di accettare queste modifiche.
4. Una volta completata la configurazione, verrà creata una nuova voce di configurazione SSH per CodeBuild sandbox.

```
Host codebuild-sandbox-ssh*
  StrictHostKeyChecking no
  LogLevel INFO
  ForwardAgent yes
  ControlMaster auto
  ControlPersist 10m
  ProxyCommand ssh -C "/Users/[redacted]/.aws/codebuild-dev-env/codebuild-sandbox-connect.sh %n"
```

Passaggio 3: Connect alla sandbox

Per modificare la configurazione SSH locale nella console

1. Configura AWS CLI l'autenticazione e assicurati che AWS CLI l'utente disponga dell'`codebuild:StartSandboxConnection` autorizzazione. Per ulteriori informazioni, consulta [Autenticazione tramite credenziali utente IAM per la Guida per l'utente dell'interfaccia AWS CLI](#) a riga di AWS comando per la versione 1.
2. Connect alla sandbox con il seguente comando:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>
```

Note

Per risolvere gli errori di connessione, usa il `-v` flag per abilitare l'output dettagliato. Ad esempio, `ssh -v codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>`.

Per ulteriori indicazioni sulla risoluzione dei problemi, vedere. [Risoluzione dei problemi AWS CodeBuild di connessione SSH nella sandbox](#)

Fase 4: Rivedi i risultati

Una volta connesso, puoi eseguire il debug degli errori di compilazione, testare i comandi di compilazione, sperimentare modifiche alla configurazione e verificare le variabili e le dipendenze di ambiente con la tua sandbox.

Risoluzione dei problemi AWS CodeBuild di connessione SSH nella sandbox

Utilizza le informazioni contenute in questo argomento per identificare, diagnosticare e risolvere i problemi di connessione SSH CodeBuild nella sandbox.

Argomenti

- [StartSandboxConnectionInvalidInputException errore quando SSH entra nell'ambiente sandbox CodeBuild](#)
- [Errore: «Impossibile individuare le credenziali» quando SSH entra nell'ambiente sandbox CodeBuild](#)
- [StartSandboxConnectionAccessDeniedException errore quando SSH entra nell'ambiente sandbox CodeBuild](#)
- [Errore: «ssh: impossibile risolvere il nome host» quando SSH entra nell'ambiente sandbox CodeBuild](#)

StartSandboxConnectionInvalidInputException errore quando SSH entra nell'ambiente sandbox CodeBuild

Problema: quando si tenta di connettersi a un ambiente CodeBuild sandbox utilizzando il comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, è possibile che si verifichi un `InvalidInputException` errore del tipo:

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection operation: Failed to start SSM session for {sandbox-arn}
User: arn:aws:sts::<account-ID>:assumed-role/<service-role-name>/AWSCodeBuild-<UUID>
is not authorized to perform: ssm:StartSession on resource.
```

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection operation: Failed to start SSM session for
sandbox <sandbox-arn>: codebuild:<UUID> is not connected.
```

Causa possibile:

- Agente Amazon EC2 Systems Manager mancante: nell'immagine di build non è installato o configurato correttamente l'agente SSM.
- Autorizzazioni insufficienti: il ruolo del servizio di CodeBuild progetto non dispone delle autorizzazioni SSM richieste.

Soluzione consigliata: se utilizzi un'immagine personalizzata per la build, procedi come segue.

1. Installazione di SSM Agent Per ulteriori informazioni, consulta [Installazione e disinstallazione manuale dell'agente SSM su istanze Amazon EC2](#) per Linux nel. La versione di SSM Agent deve essere o successiva. 3.0.1295.0
2. Copia il file, <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/7.0/amazon-ssm-agent.json> nella directory dell'immagine. /etc/amazon/ssm/ Ciò abilita la modalità contenitore nell'agente SSM.
3. Assicurati che il ruolo di servizio del tuo CodeBuild progetto disponga delle seguenti autorizzazioni, quindi riavvia l'ambiente sandbox:

```
{
  "Effect": "Allow",
  "Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
    "ssmmessages:OpenDataChannel"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:build/*",
    "arn:aws:ssm:region::document/AWS-StartSSHSession"
  ]
}
```

Errore: «Impossibile individuare le credenziali» quando SSH entra nell'ambiente sandbox CodeBuild

Problema: quando si tenta di connettersi a un ambiente CodeBuild sandbox utilizzando il comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, è possibile che si verifichi il seguente errore di credenziali:

Unable to locate credentials. You can configure credentials by running

```
"aws configure".
```

Possibile causa: AWS le credenziali non sono state configurate correttamente nell'ambiente locale.

Soluzione consigliata: configura AWS CLI le credenziali seguendo la documentazione ufficiale: [Configurazione delle impostazioni per la Guida utente dell'interfaccia AWS CLI a riga di AWS](#) comando per la versione 2.

StartSandboxConnectionAccessDeniedException errore quando SSH entra nell'ambiente sandbox CodeBuild

Problema: quando si tenta di connettersi a un ambiente CodeBuild sandbox utilizzando il comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, è possibile che si verifichi il seguente errore di autorizzazione:

```
An error occurred (AccessDeniedException) when calling the StartSandboxConnection
operation:
User: arn:aws:sts::account-id:assumed-role/role-name
is not authorized to perform: codebuild:StartSandboxConnection on resource:
sandbox-arn
because no identity-based policy allows the codebuild:StartSandboxConnection action
```

Possibile causa: AWS le tue credenziali non dispongono delle CodeBuild autorizzazioni necessarie per eseguire questa operazione.

Soluzione consigliata: assicurati che l'utente o il ruolo IAM associato alle tue AWS CLI credenziali disponga delle seguenti autorizzazioni:

```
{
  "Effect": "Allow",
  "Action": [
    "codebuild:StartSandboxConnection"
  ],
  "Resource": [
    "arn:aws:codebuild:region:account-id:sandbox/*"
  ]
}
```

Errore: «ssh: impossibile risolvere il nome host» quando SSH entra nell'ambiente sandbox CodeBuild

Problema: quando si tenta di connettersi a un ambiente CodeBuild sandbox utilizzando il comando `ssh codebuild-sandbox-ssh=<sandbox-arn>`, si verifica il seguente errore di risoluzione del nome host:

```
ssh: Could not resolve hostname
```

Possibile causa: questo errore si verifica in genere quando lo script di connessione alla CodeBuild sandbox richiesto non è stato eseguito correttamente nell'ambiente locale.

Soluzione consigliata:

1. Scarica lo script di connessione alla CodeBuild sandbox.
2. Esegui lo script nel tuo terminale per stabilire la configurazione SSH necessaria.
3. Riprova la connessione SSH all'ambiente sandbox.

Compilazioni di debug con Session Manager

In AWS CodeBuild, puoi mettere in pausa una build in esecuzione e quindi utilizzare AWS Systems Manager Session Manager per connetterti al contenitore di compilazione e visualizzare lo stato del contenitore.

Note

Questa funzionalità non è disponibile negli ambienti Windows.

Argomenti

- [Prerequisiti](#)
- [Metti in pausa la compilazione](#)
- [Avvia la compilazione](#)
- [Connect al container di compilazione](#)
- [Riprendi la compilazione](#)

Prerequisiti

Per consentire l'utilizzo di Session Manager con la sessione di compilazione, è necessario abilitare la connessione alla sessione per la build. Esistono due prerequisiti:

- CodeBuild Nelle immagini curate dagli standard Linux l'agente SSM è già installato e l'agente SSM è abilitato. ContainerMode

Se stai usando un'immagine personalizzata per la tua build, fai quanto segue:

1. Installazione di SSM Agent Per ulteriori informazioni, consulta [Installare manualmente l'agente SSM sulle istanze EC2 per Linux nella Guida per l' AWS Systems Manager utente](#). La versione di SSM Agent deve essere 3.0.1295.0 o successiva.
2. Copia il file <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent.json> nella directory dell'immagine. /etc/amazon/ssm/ Ciò abilita la modalità contenitore nell'agente SSM.

Note

Le immagini personalizzate richiederebbero l'agente SSM più aggiornato affinché questa funzionalità funzioni come previsto.

- Il ruolo CodeBuild di servizio deve avere la seguente politica SSM:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Puoi fare in modo che la CodeBuild console alleggi automaticamente questa policy al tuo ruolo di servizio all'avvio della build. In alternativa, puoi allegare manualmente questa politica al tuo ruolo di servizio.

- Se l'attività di controllo e registrazione della sessione è abilitata nelle preferenze di Systems Manager, il ruolo di CodeBuild servizio deve disporre anche di autorizzazioni aggiuntive. Le autorizzazioni sono diverse, a seconda di dove sono archiviati i log.

CloudWatch Registri

Se utilizzi CloudWatch Logs per archiviare i log, aggiungi la seguente autorizzazione al CodeBuild ruolo di servizio:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:MyLogGroup:*"
    }
  ]
}
```

Simple Storage Service (Amazon S3)

Se utilizzi Amazon S3 per archiviare i log, aggiungi la seguente autorizzazione al CodeBuild ruolo di servizio:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetEncryptionConfiguration",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Controllo e registrazione dell'attività della sessione nella Guida per l'utente AWS Systems Manager](#)

Metti in pausa la compilazione

Per mettere in pausa la compilazione, inserisci il `codebuild-breakpoint` comando in una qualsiasi delle fasi di compilazione nel tuo file `buildspec`. La compilazione verrà messa in pausa a questo punto, il che consente di connettersi al contenitore di compilazione e visualizzare il contenitore nello stato corrente.

Ad esempio, aggiungi quanto segue alle fasi di compilazione nel tuo file `buildspec`.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Questo codice crea il `/tmp/hello-world` file e quindi sospende la compilazione a questo punto.

Avvia la compilazione

Per consentire l'utilizzo di Session Manager con la sessione di compilazione, è necessario abilitare le connessioni di sessione per la build. Per fare ciò, quando avvii la build, segui questi passaggi:

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Scegli il progetto di compilazione, quindi scegli Avvia build with overrides.
3. Selezionare Advanced build overrides (Sostituzioni avanzate della compilazione).
4. Nella sezione Ambiente, scegli l'opzione Abilita la connessione alla sessione. Se questa opzione non è selezionata, tutti i codebuild-resume comandi codebuild-breakpoint and vengono ignorati.
5. Apportate le altre modifiche desiderate e scegliete Avvia compilazione.
6. Monitora lo stato della build nella console. Quando la sessione è disponibile, il link AWS Session Manager viene visualizzato nella sezione Build status.

Connect al container di compilazione

Puoi connetterti al contenitore di compilazione in due modi:

CodeBuild console

In un browser web, apri il link AWS Session Manager per connetterti al contenitore di build. Si apre una sessione di terminale che consente di sfogliare e controllare il contenitore di compilazione.

AWS CLI

Note

Per questa procedura, è necessario che sul computer locale sia installato il plug-in Session Manager. Per ulteriori informazioni, consulta [Installare il plug-in Session Manager per la AWS CLI nella Guida](#) per l' AWS Systems Manager utente.

1. Chiama l'batch-get-buildsapi con l'ID di build per ottenere informazioni sulla build, incluso l'identificatore di destinazione della sessione. Il nome della proprietà dell'identificatore di destinazione della sessione varia a seconda del tipo di output del aws comando. Questo è il motivo per cui `--output json` viene aggiunto al comando.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Copia il valore della sessionTarget proprietà. Il nome della sessionTarget proprietà può variare a seconda del tipo di output del aws comando. Questo è il motivo --output json per cui viene aggiunto al comando nel passaggio precedente.
3. Usa il seguente comando per connetterti al contenitore di compilazione.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Per questo esempio, verificate che il /tmp/hello-world file esista e contenga il testoHello World.

Riprendi la compilazione

Dopo aver finito di esaminare il contenitore di compilazione, emetti il codebuild-resume comando dalla shell del contenitore.

```
$ codebuild-resume
```

Elimina build in AWS CodeBuild

È possibile utilizzare AWS CLI o the AWS SDKs per eliminare le build. AWS CodeBuild

Argomenti

- [Eliminazione delle compilazioni \(AWS CLI\)](#)
- [Eliminazione delle compilazioni \(AWS SDKs\)](#)

Eliminazione delle compilazioni (AWS CLI)

Eseguire il comando batch-delete-builds:

```
aws codebuild batch-delete-builds --ids <ids>
```

Nel comando precedente, sostituire il seguente segnaposto:

- **ids**: stringa richiesta. Le IDs build da eliminare. Per specificare più compilazioni, separa ogni ID con uno spazio. Per ottenere un elenco delle build IDs, consulta i seguenti argomenti:
 - [Visualizza un elenco di build \(\) IDs AWS CLI](#)
 - [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#)

In caso di esito positivo, una matrice `buildsDeleted` viene visualizzata nell'output, con l'Amazon Resource Name (ARN) di ciascuna compilazione eliminata correttamente. Informazioni sulle compilazioni che non sono state eliminate correttamente sono visualizzate nell'output in una matrice `buildsNotDeleted`.

Ad esempio, se si esegue questo comando:

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Informazioni simili a quelle riportate di seguito compaiono nell'output:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
  ]
}
```

Eliminazione delle compilazioni (AWS SDKs)

Per informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Retry si integra manualmente AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs riprovare manualmente una build singola o una build in batch. AWS CLI AWS CodeBuild

Argomenti

- [Riprova una compilazione manualmente \(console\)](#)
- [Riprova una build manualmente \(\)AWS CLI](#)
- [Riprova una build manualmente \(\)AWS SDKs](#)

Riprova una compilazione manualmente (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Esegui una delle seguenti operazioni:
 - Se viene visualizzata la **build-ID** pagina **build-project-name**., scegli Riprova build.
 - Nel riquadro di navigazione, selezionare Build history (Cronologia build). Nell'elenco delle build, seleziona la casella relativa alla build, quindi scegli Riprova build.
 - Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), selezionare il collegamento del nome del progetto di compilazione. Nell'elenco delle build, seleziona la casella relativa alla build, quindi scegli Riprova build.

Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di compilazione o build più recenti. Per visualizzare altri progetti di compilazione o build, selezionare l'icona a forma di ingranaggio e selezionare un valore diverso per Builds per page (Build per pagina) o Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Riprova una build manualmente ()AWS CLI

- Eseguire il comando `retry-build`:

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

Nel comando precedente, sostituire il seguente segnaposto:

- **<build-id>**: stringa richiesta. L'ID della build o della build batch da riprovare. Per ottenere un elenco delle build IDs, consulta i seguenti argomenti:
 - [Visualizza un elenco di build \(\) IDs AWS CLI](#)
 - [Visualizza un elenco di build in batch IDs \(AWS CLI\)](#)
 - [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#)
 - [Visualizza un elenco di build in batch IDs per un progetto di compilazione \(AWS CLI\)](#)
- **--idempotency-token**: facoltativo. Se esegui il `retry-build` comando con l'opzione, nella richiesta viene incluso un identificatore o token univoco con distinzione tra maiuscole e minuscole. `retry-build` Il token è valido per 5 minuti dopo la richiesta . Se si ripete la `retry-build` richiesta con lo stesso token, ma si modifica un parametro, viene CodeBuild restituito un errore di mancata corrispondenza del parametro.

Riprova una build manualmente ()AWS SDKs

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Retry si integra automaticamente AWS CodeBuild

Puoi usare la AWS CodeBuild console o riprovare automaticamente AWS SDKs le tue build. AWS CLI AWS CodeBuild Con il ritentativo automatico abilitato, CodeBuild chiamerà automaticamente `RetryBuild` utilizzando il ruolo di servizio del progetto dopo un completamento non riuscito fino a un limite specificato. Ad esempio, se il limite di tentativi automatici è impostato su due, CodeBuild chiamerà `RetryBuildAPI` per riprovare automaticamente la build per un massimo di due volte aggiuntive.

Note

CodeBuild non supporta il ritentativo automatico per. CodePipeline

Argomenti

- [Riprova automaticamente una build \(console\)](#)
- [Riprova una build automaticamente \(\)AWS CLI](#)
- [Riprova automaticamente una build \(\)AWS SDKs](#)

Riprova automaticamente una build (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Seleziona Crea progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#) e [Esecuzione di una compilazione \(console\)](#).
 - In Environment (Ambiente):
 - Per Limite tentativi automatici, inserisci il numero massimo di tentativi automatici desiderato dopo una compilazione non riuscita.
3. In Ambiente, scegli Configurazione aggiuntiva.
4. Continua con i valori predefiniti, quindi scegli Crea progetto di compilazione.

Riprova una build automaticamente ()AWS CLI

- Eseguire il comando create-project:

```
aws codebuild create-project \
  --name "<project-name>" \
  --auto-retry-limit <auto-retry-limit> \
  --source "<source>" \
  --artifacts {<artifacts>} \
  --environment "{\"type\": \"<environment-type>\", \"image\": \"<image-type>\", \"computeType\": \"<compute-type>\"}" \
  --service-role "<service-role>"
```

Nel comando precedente, sostituire i seguenti segnaposto:

- **<auto-retry-limit>**: imposta il limite di tentativi automatici sul numero massimo di tentativi automatici desiderato dopo una compilazione non riuscita.

- `<project-name>,,<source>,<artifacts>, environment-type> image-type>compute-type>, eservice-role>`: Imposta le impostazioni di configurazione del progetto desiderate.

Riprova automaticamente una build ()AWS SDKs

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Stop si integra AWS CodeBuild

Puoi usare la AWS CodeBuild console o AWS SDKs interrompere un build in AWS CodeBuild. AWS CLI

Argomenti

- [Interruzione di una compilazione \(console\)](#)
- [Interruzione di una compilazione \(AWS CLI\)](#)
- [Interruzione di una compilazione \(AWS SDKs\)](#)

Interruzione di una compilazione (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Esegui una di queste operazioni:
 - Se viene visualizzata la **build-ID** pagina **build-project-name**:, scegli Stop build.
 - Nel riquadro di navigazione, selezionare Build history (Cronologia build). Nell'elenco di build, selezionare la casella della build e selezionare Stop build (Interrompi build).
 - Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), selezionare il collegamento del nome del progetto di compilazione. Nell'elenco di build, selezionare la casella della build e selezionare Stop build (Interrompi build).

 Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di compilazione o build più recenti. Per visualizzare altri progetti di compilazione o build, selezionare l'icona a forma di ingranaggio e selezionare un valore diverso per Builds per page (Build per pagina) o Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Se AWS CodeBuild non riesci a interrompere correttamente una compilazione (ad esempio, se il processo di compilazione è già completo), il pulsante Stop è disabilitato o potrebbe non apparire.

Interruzione di una compilazione (AWS CLI)

- Eseguire il comando stop-build:

```
aws codebuild stop-build --id id
```

Nel comando precedente, sostituire il seguente segnaposto:

- *id*: stringa obbligatoria. L'ID della compilazione da interrompere. Per ottenere un elenco delle build IDs, consulta i seguenti argomenti:
 - [Visualizza un elenco di build \(\) IDs AWS CLI](#)
 - [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#)

Se interrompe AWS CodeBuild correttamente la compilazione, il buildStatus valore nell'buildoggetto nell'output èSTOPPED.

Se CodeBuild non è possibile interrompere correttamente la compilazione (ad esempio, se la compilazione è già completa), il buildStatus valore build dell'oggetto nell'output è lo stato finale della build (ad esempio,SUCCEEDED).

Interruzione di una compilazione (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta[AWS SDKs e riferimento agli strumenti](#).

Interrompi gli accumuli in batch AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o interrompere AWS SDKs la compilazione in batch AWS CodeBuild. AWS CLI

Note

Se utilizzi il calcolo Lambda nella build in batch, la build Lambda in corso non può essere interrotta.

Argomenti

- [Interrompi una compilazione in batch \(console\)](#)
- [Interrompi una compilazione in batch \(AWS CLI\)](#)
- [Interrompi una compilazione in batch \(\)AWS SDKs](#)

Interrompi una compilazione in batch (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Esegui una di queste operazioni:
 - Se viene visualizzata la **build-ID** pagina **build-project-name**:, scegli Stop build.
 - Nel riquadro di navigazione, selezionare Build history (Cronologia build). Nell'elenco di build, selezionare la casella della build e selezionare Stop build (Interrompi build).
 - Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), selezionare il collegamento del nome del progetto di compilazione. Nell'elenco di build, selezionare la casella della build e selezionare Stop build (Interrompi build).

Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di compilazione o build più recenti. Per visualizzare altri progetti di compilazione o build, selezionare l'icona a forma di ingranaggio e selezionare un valore diverso per Builds per page (Build per pagina) o Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Interrompi una compilazione in batch (AWS CLI)

- Eseguire il comando [stop-build-batch](#):

```
aws codebuild stop-build-batch --id <batch-build-id>
```

Nel comando precedente, sostituire il seguente segnaposto:

- <batch-build-id>**: stringa richiesta. L'identificatore della build batch da interrompere. Per ottenere un elenco degli identificatori di build in batch, consultate i seguenti argomenti:
 - [Visualizza un elenco di build in batch IDs \(AWS CLI\)](#)
 - [Visualizza un elenco di build in batch IDs per un progetto di compilazione \(AWS CLI\)](#)

Interrompi una compilazione in batch (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Trigger AWS CodeBuild si crea automaticamente

Puoi creare un trigger su un progetto per pianificare una build una volta ogni ora, giorno o settimana. Puoi anche modificare un trigger per utilizzare una regola personalizzata con un'espressione CloudWatch cron di Amazon. Ad esempio, utilizzando un'espressione cron, puoi pianificare una build in un orario specifico ogni giorno feriale. Per informazioni sulla creazione e la modifica dei trigger, consulta [Creare trigger AWS CodeBuild](#) e [Modifica i AWS CodeBuild trigger](#)

Argomenti

- [Creare trigger AWS CodeBuild](#)
- [Modifica i AWS CodeBuild trigger](#)

Creare trigger AWS CodeBuild

Puoi creare un trigger su un progetto per pianificare una build una volta ogni ora, giorno o settimana. Puoi anche creare un trigger utilizzando una regola personalizzata con un'espressione CloudWatch cron di Amazon. Ad esempio, utilizzando un'espressione cron, puoi pianificare una build in un orario specifico ogni giorno feriale.

 Note

Non è possibile avviare una compilazione in batch da un trigger di compilazione, un EventBridge evento Amazon o un' AWS Step Functions attività.

Argomenti

- [Crea AWS CodeBuild trigger \(console\)](#)
- [Crea AWS CodeBuild trigger a livello di codice](#)

Crea AWS CodeBuild trigger (console)

Utilizzare la procedura seguente per creare trigger utilizzando. Console di gestione AWS

Per creare un trigger

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Selezionare il collegamento del progetto di compilazione a cui si desidera aggiungere un trigger, quindi selezionare la scheda Build triggers (Trigger della build).

 Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di build più recenti. Per visualizzare altri progetti di compilazione, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

4. Selezionare Create trigger (Crea trigger).
5. Alla voce Trigger name (Nome trigger), inserire un nome.
6. Dal menu a discesa Frequency (Frequenza), scegli la frequenza per il trigger. Se intendi creare una frequenza utilizzando un'espressione cron, seleziona Custom (Personalizzato).
7. Specifica il parametro relativo alla frequenza del tuo trigger. Puoi inserire i primi caratteri delle selezioni nella casella di testo per filtrare le voci del menu a discesa.

 Note

Le ore e i minuti di inizio sono basati su zero. Il minuto di inizio è un numero compreso tra zero e 59. L'ora di inizio è un numero compreso tra zero e 23. Ad esempio, un trigger giornaliero che inizia ogni giorno alle 12:15 ha un'ora di inizio di 12 e un minuto di inizio di 15. Un trigger giornaliero che inizia ogni giorno a mezzanotte ha un'ora di inizio pari a zero e un minuto di inizio pari a zero. Un trigger giornaliero che inizia ogni giorno alle 23:59 ha un'ora di inizio di 23 e un minuto di inizio di 59.

Frequenza	Parametri obbligatori	Informazioni
Orario	Avvia minuto	Utilizza il menu a discesa Start minute (Avvia minuto).
Giornaliero	Avvia minuto	Utilizza il menu a discesa Start minute (Avvia minuto).
	Avvia ora	Utilizza il menu a discesa Start hour (Avvia ora).
Settimanale	Avvia minuto	Utilizza il menu a discesa Start minute (Avvia minuto).
	Avvia ora	Utilizza il menu a discesa Start hour (Avvia ora).
	Avvia giorno	Utilizza il menu a discesa Start day (Avvia giorno).

Frequenza	Parametri obbligatori	Informazioni
Personalizza	Espressione cron	Inserisci un'espressione cron in Cron expression (Espressione cron). Un'espressione cron ha sei campi obbligatori separati da uno spazio bianco. I campi specificano un valore di avvio per minuti, ore, giorno del mese, mese, giorno della settimana e anno. Puoi usare i caratteri jolly per specificare un range, valori aggiuntivi e altro ancora. Ad esempio, l'espressione cron 0 9 ? * MON-FRI * pianifica una build ogni giorno della settimana alle 9:00. Per ulteriori informazioni, consulta Cron Expressions nella Amazon Events User Guide. CloudWatch

8. Seleziona Enable this trigger (Attiva questo trigger).
9. (Facoltativo) Espandere Advanced section (Sezione avanzata). Digitare una versione della sorgente in Source version (Versione sorgente).
 - Per Amazon S3, inserisci l'ID della versione che corrisponde alla versione dell'elemento di input che desideri creare. Se il valore Source version (Versione sorgente) non è specificato, viene utilizzata l'ultima versione.
 - Per AWS CodeCommit, digita un ID di commit. Se Source version (Versione di origine) non è specificata, viene utilizzato l'ID commit HEAD del ramo predefinito.
 - Per GitHub o GitHub Enterprise, digita un ID di commit, un ID di pull request, un nome di filiale o un nome di tag che corrisponda alla versione del codice sorgente che desideri creare. Se specifichi l'ID di una richiesta pull, devi utilizzare il formato pr/*pull-request-ID* (ad esempio, pr/25). Se specifichi un nome ramo, viene utilizzato l'ID commit HEAD del ramo. Se

Source version (Versione sorgente) non è specificata, viene utilizzato l'ID commit HEAD del ramo predefinito.

- Per Bitbucket, digita un ID commit, un nome ramo o un nome tag che corrispondono alla versione del codice sorgente che desideri utilizzare. Se specifichi un nome ramo, viene utilizzato l'ID commit HEAD del ramo. Se Source version (Versione sorgente) non è specificata, viene utilizzato l'ID commit HEAD del ramo predefinito.
10. (Facoltativo) Specificate un timeout compreso tra 5 minuti e 2160 minuti (36 ore). Questo valore specifica per quanto tempo AWS CodeBuild tenta una compilazione prima che si interrompa. Se Hours (Ore) e Minutes (Minuti) vengono lasciati vuoti, viene utilizzato il valore del timeout predefinito specificato nel progetto.
 11. Selezionare Create trigger (Crea trigger).

Crea AWS CodeBuild trigger a livello di codice

CodeBuild utilizza EventBridge le regole di Amazon per i trigger di compilazione. Puoi utilizzare l' EventBridge API per creare programmaticamente trigger di compilazione per i tuoi progetti.

CodeBuild Per ulteriori informazioni, consulta [Amazon EventBridge API Reference](#).

Modifica i AWS CodeBuild trigger

Puoi modificare un trigger su un progetto per pianificare una build una volta ogni ora, giorno o settimana. Puoi anche modificare un trigger per utilizzare una regola personalizzata con un'espressione CloudWatch cron di Amazon. Ad esempio, utilizzando un'espressione cron, puoi pianificare una build in un orario specifico ogni giorno feriale. Per informazioni su come creare un trigger , vedi [Creare trigger AWS CodeBuild](#).

Argomenti

- [Modifica i AWS CodeBuild trigger \(console\)](#)
- [Modifica i AWS CodeBuild trigger a livello di codice](#)

Modifica i AWS CodeBuild trigger (console)

Utilizzare la procedura seguente per modificare i trigger utilizzando. Console di gestione AWS

Per modificare un trigger

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Selezionare il collegamento del progetto di compilazione da modificare, quindi selezionare la scheda Build triggers (Trigger della build).

Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di build più recenti. Per visualizzare altri progetti di compilazione, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

4. Selezionare il pulsante di opzione accanto al trigger da modificare, quindi selezionare Edit (Modifica).
5. Dal menu a discesa Frequency (Frequenza), scegli la frequenza per il trigger. Se intendi creare una frequenza utilizzando un'espressione cron, seleziona Custom (Personalizzato).
6. Specifica il parametro relativo alla frequenza del tuo trigger. Puoi inserire i primi caratteri delle selezioni nella casella di testo per filtrare le voci del menu a discesa.

Note

Le ore e i minuti di inizio sono basati su zero. Il minuto di inizio è un numero compreso tra zero e 59. L'ora di inizio è un numero compreso tra zero e 23. Ad esempio, un trigger giornaliero che inizia ogni giorno alle 12:15 ha un'ora di inizio di 12 e un minuto di inizio di 15. Un trigger giornaliero che inizia ogni giorno a mezzanotte ha un'ora di inizio pari a zero e un minuto di inizio pari a zero. Un trigger giornaliero che inizia ogni giorno alle 23:59 ha un'ora di inizio di 23 e un minuto di inizio di 59.

Frequenza	Parametri obbligatori	Informazioni
Orario	Avvia minuto	Utilizza il menu a discesa Start minute (Avvia minuto).
Giornaliero	Avvia minuto Avvia ora	Utilizza il menu a discesa Start minute (Avvia minuto).

Frequenza	Parametri obbligatori	Informazioni
		Utilizza il menu a discesa Start hour (Avvia ora).
Settimanale	Avvia minuto	Utilizza il menu a discesa Start minute (Avvia minuto).
	Avvia ora	
	Avvia giorno	Utilizza il menu a discesa Start hour (Avvia ora).
		Utilizza il menu a discesa Start day (Avvia giorno).
Personalizza	Espressione cron	Inserisci un'espressione cron in Cron expression (Espressione cron). Un'espressione cron ha sei campi obbligatori separati da uno spazio bianco. I campi specificano un valore di avvio per minuti, ore, giorno del mese, mese, giorno della settimana e anno. Puoi usare i caratteri jolly per specificare un range, valori aggiuntivi e altro ancora. Ad esempio, l'espressione cron 0 9 ? * MON-FRI * pianifica una build ogni giorno della settimana alle 9:00. Per ulteriori informazioni, consulta Cron Expressions nella Amazon Events User Guide. CloudWatch

7. Seleziona Enable this trigger (Attiva questo trigger).

 Note

Puoi utilizzare la CloudWatch console Amazon all'indirizzo <https://console.aws.amazon.com/cloudwatch/> per modificare la versione sorgente, il timeout e altre opzioni non disponibili in AWS CodeBuild.

Modifica i AWS CodeBuild trigger a livello di codice

CodeBuild utilizza EventBridge le regole di Amazon per i trigger di compilazione. Puoi utilizzare l' EventBridge API per modificare a livello di codice i trigger di compilazione per i tuoi progetti. CodeBuild Per ulteriori informazioni, consulta [Amazon EventBridge API Reference](#).

Visualizza i dettagli della costruzione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o visualizzare AWS SDKs i dettagli sulle build gestite da. AWS CLI CodeBuild

Argomenti

- [Visualizzazione dei dettagli delle compilazioni \(console\)](#)
- [Visualizzazione dei dettagli delle compilazioni \(AWS CLI\)](#)
- [Visualizzazione dei dettagli delle compilazioni \(AWS SDKs\)](#)
- [Transizioni delle fasi della compilazione](#)

Visualizzazione dei dettagli delle compilazioni (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Esegui una delle seguenti operazioni:
 - Nel riquadro di navigazione, selezionare Build history (Cronologia build). Nell'elenco delle build, nella colonna Build run (Esecuzione build), selezionare il collegamento della build.
 - Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), selezionare il collegamento del nome del progetto di compilazione. Nell'elenco delle build, nella colonna Build run (Esecuzione build), selezionare il collegamento della build.

 Note

Per impostazione predefinita, vengono visualizzati i 10 progetti di compilazione o build più recenti. Per visualizzare altri progetti di compilazione o build, selezionare l'icona a forma di ingranaggio e selezionare un valore diverso per Builds per page (Build per pagina) o Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Visualizzazione dei dettagli delle compilazioni (AWS CLI)

Per ulteriori informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta. [Guida di riferimento alla riga di comando](#)

Eseguire il comando batch-get-builds:

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

Sostituire il seguente segnaposto:

- *ids*: stringa richiesta. Una o più build IDs per visualizzarne i dettagli. Per specificare più di un ID build, separare ogni ID con uno spazio. Puoi specificare fino a 100 build IDs. Per ottenere un elenco delle build IDs, consulta i seguenti argomenti:
 - [Visualizza un elenco di build \(\) IDs AWS CLI](#)
 - [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#)

Ad esempio, se si esegue questo comando:

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

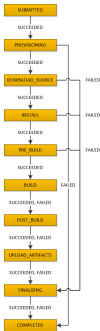
In caso di esito positivo, nell'output vengono visualizzati dati simili a quelli descritti nella procedura [Per visualizzare un riepilogo delle informazioni sulla compilazione](#).

Visualizzazione dei dettagli delle compilazioni (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Transizioni delle fasi della compilazione

Le build AWS CodeBuild procedono per fasi:



 Important

La fase `UPLOAD_ARTIFACTS` viene sempre tentata, anche se la fase `BUILD` fallisce.

Visualizza un elenco di build IDs in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs visualizzare un elenco di build IDs per le build gestite da CodeBuild. AWS CLI

Argomenti

- [Visualizza un elenco di build IDs \(console\)](#)
- [Visualizza un elenco di build \(\) IDs AWS CLI](#)
- [Visualizza un elenco di build in batch IDs \(AWS CLI\)](#)
- [Visualizza un elenco di build IDs \(AWS SDKs\)](#)

Visualizza un elenco di build IDs (console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Nel riquadro di navigazione, selezionare Build history (Cronologia build).

Note

Per impostazione predefinita, vengono visualizzate solo le 10 build più recenti. Per visualizzare altre build, selezionare l'icona a forma di ingranaggio, quindi selezionare un valore diverso per Builds per page (Build per pagina) oppure utilizzare le frecce avanti e indietro.

Visualizza un elenco di build () IDs AWS CLI

Per ulteriori informazioni sull'utilizzo di AWS CLI with CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

- Eseguire il comando list-builds:

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

Nel comando precedente, sostituire i seguenti segnaposto:

- *sort-order*: Stringa opzionale utilizzata per indicare come elencare la build IDs. I valori validi includono ASCENDING e DESCENDING.
- *next-token*: stringa opzionale. Se durante un'esecuzione precedente c'erano più di 100 voci nell'elenco, vengono restituiti solo i primi 100 elementi, insieme a una stringa univoca denominata token successivo. Per ottenere il gruppo successivo di elementi nell'elenco, eseguire il comando nuovamente, aggiungendo il token successivo alla chiamata. Per ottenere tutti gli elementi dell'elenco, è necessario continuare a eseguire il comando con ogni token successivo restituito, finché non terminano i token successivi restituiti.

Ad esempio, se si esegue questo comando:

```
aws codebuild list-builds --sort-order ASCENDING
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
```

```
"nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
"ids": [
  "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
  "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE",
  ... The full list of build IDs has been omitted for brevity ...
  "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}
```

Se si esegue nuovamente il comando:

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full
token has been omitted for brevity...MzY2OA==
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Visualizza un elenco di build in batch IDs (AWS CLI)

Per ulteriori informazioni sull'utilizzo di AWS CLI with CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

- Eseguire il comando list-build-batches:

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

Nel comando precedente, sostituire i seguenti segnaposto:

- sort-order*: Stringa facoltativa utilizzata per indicare come elencare la build in batch IDs. I valori validi includono ASCENDING e DESCENDING.

- *next-token*: stringa opzionale. Se durante un'esecuzione precedente c'erano più di 100 voci nell'elenco, vengono restituiti solo i primi 100 elementi, insieme a una stringa univoca denominata token successivo. Per ottenere il gruppo successivo di elementi nell'elenco, eseguire il comando nuovamente, aggiungendo il token successivo alla chiamata. Per ottenere tutti gli elementi dell'elenco, è necessario continuare a eseguire il comando con ogni token successivo restituito, finché non terminano i token successivi restituiti.

Ad esempio, se si esegue questo comando:

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Se si esegue nuovamente il comando:

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The
full token has been omitted for brevity...MzY20A==
```

Nell'output potrebbe essere visualizzato un risultato come il seguente:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Visualizza un elenco di build IDs (AWS SDKs)

Per ulteriori informazioni sull'utilizzo CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Visualizza un elenco di build IDs per un progetto di compilazione in AWS CodeBuild

È possibile utilizzare la AWS CodeBuild console o AWS SDKs visualizzare un elenco di build IDs per un progetto di compilazione in CodeBuild. AWS CLI

Argomenti

- [Visualizza un elenco di build IDs per un progetto di compilazione \(console\)](#)
- [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#)
- [Visualizza un elenco di build in batch IDs per un progetto di compilazione \(AWS CLI\)](#)
- [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS SDKs\)](#)

Visualizza un elenco di build IDs per un progetto di compilazione (console)

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione). Nell'elenco dei progetti di compilazione, nella colonna Name (Nome), scegliere il progetto di compilazione.

Note

Per impostazione predefinita, vengono visualizzati i 100 progetti di compilazione o build più recenti. Per visualizzare altri progetti di compilazione o build, selezionare l'icona a forma di ingranaggio e selezionare un valore diverso per Builds per page (Build per pagina) o Projects per page (Progetti per pagina) oppure utilizzare le frecce avanti e indietro.

Visualizza un elenco di build IDs per un progetto di compilazione (AWS CLI)

Per ulteriori informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

Eseguire il comando `list-builds-for-project`, come segue:

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Nel comando precedente, sostituire i seguenti segnaposto:

- *project-name*: Stringa obbligatoria utilizzata per indicare il nome del progetto di build per cui elencare le build IDs. Per ottenere un elenco di progetti di compilazione, consulta [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#).
- *sort-order*: stringa opzionale utilizzata per indicare come elencare la build IDs. I valori validi includono ASCENDING e DESCENDING.
- *next-token*: stringa opzionale. Se durante un'esecuzione precedente c'erano più di 100 voci nell'elenco, vengono restituiti solo i primi 100 elementi, insieme a una stringa univoca denominata token successivo. Per ottenere il gruppo successivo di elementi nell'elenco, eseguire il comando nuovamente, aggiungendo il token successivo alla chiamata. Per ottenere tutte le voci nell'elenco, è necessario continuare a eseguire il comando con ogni token successivo restituito, finché non terminano i token successivi restituiti.

Ad esempio, se si esegue un comando simile a questo:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Nell'output potrebbe comparire un risultato come il seguente:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Se si esegue nuovamente il comando:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --
sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY2OA==
```

Nell'output potrebbe comparire un risultato come il seguente:

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Visualizza un elenco di build in batch IDs per un progetto di compilazione (AWS CLI)

Per ulteriori informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

Eseguire il comando `list-build-batches-for-project`, come segue:

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-
order sort-order --next-token next-token
```

Nel comando precedente, sostituire i seguenti segnaposto:

- *project-name*: Stringa obbligatoria utilizzata per indicare il nome del progetto di build per cui elencare le build IDs. Per ottenere un elenco di progetti di compilazione, consulta [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#).
- *sort-order*: stringa opzionale utilizzata per indicare come elencare la build IDs. I valori validi includono ASCENDING e DESCENDING.
- *next-token*: stringa opzionale. Se durante un'esecuzione precedente c'erano più di 100 voci nell'elenco, vengono restituiti solo i primi 100 elementi, insieme a una stringa univoca denominata token successivo. Per ottenere il gruppo successivo di elementi nell'elenco, eseguire il comando nuovamente, aggiungendo il token successivo alla chiamata. Per ottenere tutte le voci nell'elenco, è necessario continuare a eseguire il comando con ogni token successivo restituito, finché non terminano i token successivi restituiti.

Ad esempio, se si esegue un comando simile a questo:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Nell'output potrebbe comparire un risultato come il seguente:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Se si esegue nuovamente il comando:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project
--sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY20A==
```

Nell'output potrebbe comparire un risultato come il seguente:

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Visualizza un elenco di build IDs per un progetto di compilazione (AWS SDKs)

Per ulteriori informazioni sull'utilizzo AWS CodeBuild con AWS SDKs, consulta [AWS SDKs e riferimento agli strumenti](#).

Rapporti di prova in AWS CodeBuild

È possibile creare report CodeBuild contenenti dettagli sui test eseguiti durante le compilazioni. Si possono effettuare test unitari, di configurazione e funzionali.

Sono supportati i seguenti formati di file di report di test:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNg XML (.xml)
- Visual Studio TRX (.trx)
- Codice XML di Visual Studio TRX (.xml)

Note

L'ultima versione supportata di `cucumber-js` è la 7.3.2.

Crea i tuoi casi di test con qualsiasi framework di test in grado di creare file di report in uno di questi formati (ad esempio, il JUnit plug-in Surefire, TestNG o Cucumber).

Per creare un report di test, bisogna aggiungere un nome di gruppo di report al file `buildspec` di un progetto di compilazioni, con le informazioni sui casi di test. In concomitanza con il progetto di compilazione, vengono eseguiti i casi di test e viene creato un report di test. Un nuovo rapporto di test viene creato nel gruppo di report ogni volta che vengono eseguiti i casi di test. Non è necessario creare un gruppo di report prima di eseguire i test. Se specifichi il nome di un gruppo di report, CodeBuild crea automaticamente un gruppo di report quando esegui i report. Per utilizzare un gruppo di report già esistente, basta specificarne l'ARN nel file `buildspec`.

È possibile utilizzare un report di test per risolvere un problema durante l'esecuzione di una compilazione. Se si dispone di numerosi report di test, derivanti da più compilazioni di un progetto, è possibile utilizzarli per rilevare le tendenze e le percentuali di test ed errori, al fine di ottimizzare le compilazioni.

Un report scade 30 giorni dopo la sua creazione. Non è possibile visualizzare un report di test scaduto. Se desideri conservare i report dei test per più di 30 giorni, puoi esportare i file di dati grezzi dei risultati dei test in un bucket Amazon S3. I file di test esportati non scadono. Le informazioni sul bucket S3 vengono specificate quando si crea il gruppo di report.

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni di caricamento nel bucket S3.

Argomenti

- [Crea report di test](#)
- [Crea report sulla copertura del codice](#)
- [Individuazione automatica dei report in CodeBuild](#)
- [Gruppi di report](#)
- [Framework di test](#)
- [Visualizzazione dei report di test](#)
- [Autorizzazioni per il rapporto di prova](#)
- [Stati del rapporto di test](#)

Crea report di test

Per creare un report di test, è possibile eseguire un progetto di compilazione configurato con uno a cinque gruppi di report nel relativo file buildspec. Durante l'esecuzione, viene creato un report di test, contenente i risultati dei casi di test specificati per i gruppi di report. Per ogni compilazione successiva che dovesse utilizzare lo stesso file buildspec, verrà generato un nuovo report di test.

Come creare un report di test

1. Creare un progetto di compilazione. Per informazioni, consulta [Crea un progetto di compilazione in AWS CodeBuild](#).
2. Configurare il file buildspec del progetto con le informazioni relative al report di test:
 - a. Aggiungere una `reports`: sezione e specificare l'ARN di un gruppo di report esistente o il nome di un gruppo di report.

Se si specifica un ARN, CodeBuild utilizza quel gruppo di report.

Se specificate un nome, CodeBuild crea automaticamente un gruppo di rapporti utilizzando il nome del progetto e il nome specificato, nel formato `<project-name> -<report-group-name>`. Se il gruppo di report denominato esiste già, CodeBuild utilizza quel gruppo di report.

- b. Nel gruppo di report, specifica la posizione dei file che contengono i risultati del test. Se si utilizzano più gruppi di report, specificare le posizioni dei file con i risultati dei test per ciascuno di essi. A ogni esecuzione del progetto di compilazione corrisponde la creazione di un nuovo report di test. Per ulteriori informazioni, consulta [Specifica dei file di test](#).
- c. Nella sezione `commands` della sequenza `build` o `post_build`, indicare i comandi che eseguono i casi di test specificati per i gruppi di report. Per ulteriori informazioni, consulta [Specifica dei comandi di test](#).

Di seguito è riportato un esempio di sezione `buildspecreports`:

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Eseguire una compilazione del progetto. Per ulteriori informazioni, consulta [Esegui AWS CodeBuild le build manualmente](#).
4. Al termine, selezionare la nuova esecuzione di compilazione da Build history (Cronologia delle compilazioni) alla pagina del progetto. Scegliere Reports (Report) per visualizzare il report di test. Per ulteriori informazioni, consulta [Visualizzazione dei report di test in una compilazione](#).

Crea report sulla copertura del codice

CodeBuild consente di generare report sulla copertura del codice per i test. Vengono forniti i seguenti rapporti sulla copertura del codice:

Copertura della linea

La copertura della linea misura il numero di dichiarazioni coperte dai test. Una dichiarazione è una singola istruzione, esclusi commenti o condizioni.

```
line coverage = (total lines covered)/(total number of lines)
```

Copertura delle filiali

La copertura delle filiali misura il numero di filiali coperte dai test su ogni possibile ramo di una struttura di controllo, ad esempio una case dichiarazione `if` o.

```
branch coverage = (total branches covered)/(total number of branches)
```

Sono supportati i seguenti formati di file di report sulla copertura del codice:

- JaCoCo XML
- SimpleCov JSON¹
- Trifoglio XML
- Copertura XML
- INFORMAZIONI LCOV

¹ [CodeBuild accetta i report sulla copertura del codice JSON generati da simplecov, non da simplecov-json.](#)

Crea un rapporto sulla copertura del codice

Per creare un rapporto sulla copertura del codice, esegui un progetto di compilazione configurato con almeno un gruppo di rapporti sulla copertura del codice nel relativo file `buildspec`. CodeBuild interpreterà i risultati della copertura del codice e fornirà un rapporto sulla copertura del codice per l'esecuzione. Per ogni compilazione successiva che dovesse utilizzare lo stesso file `buildspec`, verrà generato un nuovo report di test.

Come creare un report di test

1. Creare un progetto di compilazione. Per informazioni, consulta [Crea un progetto di compilazione in AWS CodeBuild](#).
2. Configura il file `buildspec` del tuo progetto con le informazioni del rapporto di test:

- a. Aggiungi una `reports`: sezione e specifica il nome per il tuo gruppo di report. CodeBuild crea automaticamente un gruppo di report utilizzando il nome del progetto e il nome specificato nel formato `project-name-report-group-name-in-buildspec`. Se si dispone già di un gruppo di report da utilizzare, basta specificarne l'ARN. Se si utilizza il nome anziché l'ARN, CodeBuild crea un nuovo gruppo di report. Per ulteriori informazioni, consulta [Reports syntax in the buildspec file](#).
- b. Nel gruppo di report, specificate la posizione dei file che contengono i risultati della copertura del codice. Se utilizzi più di un gruppo di report, specifica le posizioni dei file dei risultati per ogni gruppo di report. Ogni volta che viene eseguito il progetto di compilazione viene creato un nuovo rapporto sulla copertura del codice. Per ulteriori informazioni, consulta [Specifica dei file di test](#).

Questo è un esempio che genera un rapporto sulla copertura del codice per un file di risultati JaCoCo XML che si trova in `test-results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

- c. Nella `commands` sezione della `post_build` sequenza `build or`, specificate i comandi che eseguono l'analisi della copertura del codice. Per ulteriori informazioni, consulta [Specifica dei comandi di test](#).
3. Eseguire una compilazione del progetto. Per ulteriori informazioni, consulta [Esegui AWS CodeBuild le build manualmente](#).
4. Al termine, selezionare la nuova esecuzione di compilazione da Build history (Cronologia delle compilazioni) alla pagina del progetto. Scegliete Rapporti per visualizzare il rapporto sulla copertura del codice. Per ulteriori informazioni, consulta [Visualizzazione dei report di test in una compilazione](#).

Individuazione automatica dei report in CodeBuild

Con l'individuazione automatica, CodeBuild cerca in tutti i file di build una volta completata la fase di compilazione, cerca tutti i tipi di file di report supportati e crea automaticamente nuovi gruppi e report

di test e copertura del codice. Per tutti i tipi di report rilevati, CodeBuild crea nuovi gruppi di report con lo schema seguente:

```
<project-name>-<report-file-format>-AutoDiscovered
```

Note

Se i file di rapporto rilevati hanno lo stesso tipo di formato, verranno inseriti nello stesso gruppo di rapporti o report.

L'individuazione automatica dei report è configurata in base alle variabili di ambiente del progetto:

CODEBUILD_CONFIG_AUTO_DISCOVER

Questa variabile determina se l'individuazione automatica del report è disabilitata durante la compilazione. Per impostazione predefinita, l'individuazione automatica dei report è abilitata per tutte le build. Per disabilitare questa funzionalità, imposta su `CODEBUILD_CONFIG_AUTO_DISCOVER`. `false`

CODEBUILD_CONFIG_AUTO_DISCOVER_DIR

(Facoltativo) Questa variabile determina dove vengono CodeBuild cercati i potenziali file di report. Tieni presente che, per impostazione predefinita, CodeBuild esegue la ricerca in `**/*`.

Queste variabili di ambiente possono essere modificate durante la fase di compilazione. Ad esempio, se desideri abilitare l'individuazione automatica dei report solo per le build sul ramo `main` git, puoi controllare il ramo git durante il processo di compilazione e `CODEBUILD_CONFIG_AUTO_DISCOVER` impostarlo su `false` se la build non è sul `main` ramo. L'individuazione automatica dei report può essere disabilitata utilizzando la console o utilizzando le variabili di ambiente del progetto.

Argomenti

- [Configura l'individuazione automatica dei report utilizzando la console](#)
- [Configura l'individuazione automatica dei report utilizzando le variabili di ambiente del progetto](#)

Configura l'individuazione automatica dei report utilizzando la console

Utilizzare la procedura seguente per configurare l'individuazione automatica dei report tramite la console.

Per configurare l'individuazione automatica dei report tramite la console

1. Crea un progetto di compilazione o scegli un progetto di compilazione da modificare. Per informazioni, consulta [Crea un progetto di compilazione in AWS CodeBuild](#) or [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).
2. In Ambiente, seleziona Configurazione aggiuntiva.
3. Per disabilitare l'individuazione automatica dei report, in Individuazione automatica dei report, seleziona Disattiva l'individuazione automatica dei report.
4. (Facoltativo) Nella directory di individuazione automatica: facoltativo, inserisci uno schema di directory per CodeBuild cercare i file in formato di report supportati. Tieni presente che per CodeBuild impostazione predefinita esegue la ricerca in `**/*`.

Configura l'individuazione automatica dei report utilizzando le variabili di ambiente del progetto

Utilizzare la procedura seguente per configurare l'individuazione automatica dei report utilizzando le variabili di ambiente del progetto.

Per configurare l'individuazione automatica dei report utilizzando le variabili di ambiente del progetto

1. Crea un progetto di compilazione o scegli un progetto di compilazione da modificare. Per informazioni, consulta [Crea un progetto di compilazione in AWS CodeBuild](#) or [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).
2. In Variabili di ambiente, procedi come segue:
 - a. Per disabilitare l'individuazione automatica del report, per Nome, immettere **CODEBUILD_CONFIG_AUTO_DISCOVER** e per Valore, immettere **false**. Ciò disabilita l'individuazione automatica dei report.
 - b. (Facoltativo) In Nome, invio **CODEBUILD_CONFIG_AUTO_DISCOVER_DIR** e Valore, immettete la directory in cui CodeBuild cercare i file in formato di report supportati. Ad esempio, `output/*.xml` cerca `.xml` i file nella directory `output`

Gruppi di report

Un gruppo di report contiene i report di test e specifica le impostazioni condivise. Il file `buildspec` consente di specificare i casi di test con i comandi da eseguire al momento della compilazione.

Un'esecuzione del progetto di compilazione crea un report di test per ogni gruppo di report in esso configurato. Più esecuzioni di un progetto di compilazione configurato con un gruppo di report creano più report di test nello stesso gruppo, ciascuno contenente i risultati dei casi di test specificati.

I casi di test afferenti a un gruppo di report sono indicati nel file `buildspec` di un progetto di compilazione. È possibile specificare fino a cinque gruppi di report, in un progetto di compilazione. In concomitanza con l'avvio di una compilazione, vengono eseguiti tutti i casi di test. Per ogni caso di test associato a un gruppo di report viene creato un nuovo report di test con i risultati. Ogni volta che si effettua una nuova compilazione, vengono eseguiti tutti i casi di test, i cui risultati vanno a generare un inedito report di test.

I gruppi di report possono essere utilizzati in più di un progetto di compilazione. Tutti i report di test creati con un gruppo di report condividono la stessa configurazione, ad esempio l'opzione di esportazione e le autorizzazioni, anche se generati da progetti di compilazione diversi. I report di test creati con un gruppo di report in più progetti di compilazione possono contenere i risultati derivanti dall'esecuzione di vari set di casi di test (un set per ogni progetto di compilazione). Questo perché è possibile specificare diversi file di casi di test per il gruppo di report nel file `buildspec` di ogni progetto. È inoltre possibile modificare i file dei casi di test per un gruppo di report in un progetto di compilazione, aggiornandone il file `buildspec`. Le esecuzioni di compilazione successive creeranno, quindi, nuovi report di test contenenti i risultati relativi ai file dei casi di test nel `buildspec` aggiornato.

Argomenti

- [Creazione di un gruppo di report](#)
- [Denominazione dei gruppi di report](#)
- [Condividi gruppi di report](#)
- [Specifica dei file di test](#)
- [Specifica dei comandi di test](#)
- [Aggiungi tag a un gruppo di report in AWS CodeBuild](#)
- [Aggiornamento di un gruppo di report](#)

Creazione di un gruppo di report

È possibile utilizzare la CodeBuild console AWS CLI, il o un file buildspec per creare un gruppo di report. Il ruolo IAM deve disporre delle autorizzazioni necessarie per creare un gruppo di report. Per ulteriori informazioni, consulta [Autorizzazioni per il rapporto di prova](#).

Argomenti

- [Creazione di un gruppo di report \(buildspec\)](#)
- [Creazione di un gruppo di report \(console\)](#)
- [Creazione di un gruppo di report \(CLI\)](#)
- [Creazione di un gruppo di report \(CloudFormation\)](#)

Creazione di un gruppo di report (buildspec)

Un gruppo di report creato utilizzando buildspec non esporta i file dei risultati dei test non elaborati. È possibile visualizzare il gruppo di report e specificare le impostazioni di esportazione. Per ulteriori informazioni, consulta [Aggiornamento di un gruppo di report](#).

Come creare un gruppo di report utilizzando un file buildspec

1. Scegli un nome di gruppo di rapporti che non sia associato a un gruppo di rapporti nel tuo account. AWS
2. Configurare la sezione `reports` del file buildspec con tale nome. In questo esempio, il nome del gruppo di report è `new-report-group` e i casi di test d'uso vengono creati con il JUnit framework:

```
reports:
  new-report-group: #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Il nome del gruppo di report può anche essere specificato utilizzando le variabili di ambiente nel buildspec:

```
version: 0.2
env:
  variables:
```

```
REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
  ...
reports:
  $REPORT_GROUP_NAME:
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Per ulteriori informazioni, consultare [Specifica dei file di test](#) e [Reports syntax in the buildspec file](#).

3. Nella sezione commands, specificare il comando di esecuzione dei test. Per ulteriori informazioni, consulta [Specifica dei comandi di test](#).
4. Eseguire la compilazione. Al termine della compilazione, viene creato un nuovo gruppo di report con un nome che utilizza il formato project-name-report-group-name. Per ulteriori informazioni, consulta [Denominazione dei gruppi di report](#).

Creazione di un gruppo di report (console)

Utilizzare la procedura seguente per creare un gruppo di report utilizzando. Console di gestione AWS

Per creare un gruppo di report

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Report groups (Gruppi di report).
3. Selezionare Create report group (Crea gruppo di report).
4. Immettere un nome per il proprio gruppo di report in Report group name (Nome gruppo di report).
5. (Facoltativo) Per i tag, inserisci il nome e il valore di tutti i tag che desideri vengano utilizzati dai AWS servizi di supporto. Utilizzare Add row (Aggiungi riga) per aggiungere un tag. Puoi aggiungere fino a 50 tag.
6. Se desideri caricare i dati grezzi dei risultati del rapporto di test in un bucket Amazon S3:
 - a. Seleziona Esporta in Amazon S3.
 - b. Immettere il nome del proprio bucket S3 in S3 bucket name (Nome bucket S3).

- c. (Facoltativo) Per il proprietario del bucket S3, inserisci l'identificatore dell' AWS account che possiede il bucket S3. In questo modo i dati dei report possono essere esportati in un bucket Amazon S3 di proprietà di un account diverso dall'account che esegue la compilazione.
- d. Per Path prefix (Prefisso percorso), immettere il percorso di caricamento dei risultati del test nel bucket S3.
- e. Per comprimere i file di dati non elaborati con i risultati del test, selezionare Compress test result data in a zip file (Comprimi risultati test in file zip).
- f. Per visualizzare le opzioni di crittografia, espandere Additional configuration (Configurazione aggiuntiva). Seleziona una delle seguenti opzioni:
 - Chiave AWS gestita predefinita da utilizzare Chiave gestita da AWS per Amazon S3. Per ulteriori informazioni, consulta [Customer managed CMKs](#) in the AWS Key Management Service User Guide. Questa è l'opzione di crittografia predefinita.
 - Scegli una chiave personalizzata per utilizzare una chiave gestita dal cliente che crei e configuri. In AWS KMS encryption key (Chiave di crittografia), immettere l'ARN della chiave di crittografia. Il formato è: `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Per ulteriori informazioni, consulta [Creazione di chiavi KMS](#) nella Guida per l'utente di AWS Key Management Service .
 - Disable artifact encryption (Disabilita crittografia artefatti) consente di disabilitare la crittografia. Bisogna scegliere questa opzione se si desidera condividere i risultati dei test o pubblicarli su un sito Web statico. (Un sito Web dinamico può eseguire codice per decrittare i risultati dei test.)

Per ulteriori informazioni sulla crittografia dei dati inattivi, consulta [Crittografia dei dati](#).

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni di caricamento nel bucket S3.

7. Selezionare Create report group (Crea gruppo di report).

Creazione di un gruppo di report (CLI)

Utilizza la procedura seguente per creare un gruppo di report utilizzando. AWS CLI

Per creare un gruppo di report

1. Crea un file denominato `CreateReportGroup.json`.
2. A seconda delle esigenze, copiare uno dei seguenti frammenti di codice JSON in `CreateReportGroup.json`:
 - Utilizza il seguente codice JSON per specificare che il gruppo di report di test esporta i file dei risultati dei test non elaborati in un bucket Amazon S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    },
    "tags": [
      {
        "key": "tag-key",
        "value": "tag-value"
      }
    ]
  }
}
```

- Sostituiscilo `<bucket-name>` con il nome del bucket Amazon S3 e `<path>` con il percorso nel bucket verso cui desideri esportare i file.
- Se si desidera comprimere i file esportati, in `packaging` specificare ZIP. In caso contrario, specificare NONE.
- `bucketOwner` è facoltativo ed è richiesto solo se il bucket Amazon S3 è di proprietà di un account diverso dall'account che esegue la build.
- Utilizzare `encryptionDisabled` per stabilire se crittografare i file esportati. Se crittografi i file esportati, inserisci la chiave gestita dal cliente. Per ulteriori informazioni, consulta [Aggiornamento di un gruppo di report](#).

- Il seguente JSON consente di specificare che il report di test non esporti file di test grezzi:

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni di caricamento nel bucket S3.

3. Esegui il comando seguente:

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

Creazione di un gruppo di report (CloudFormation)

Utilizza le seguenti istruzioni per creare un gruppo di report utilizzando il modello CloudFormation

Per creare un gruppo di report utilizzando il CloudFormation modello

È possibile utilizzare un file CloudFormation modello per creare ed eseguire il provisioning di un gruppo di report. Per ulteriori informazioni, consulta la [Guida per l'utente di CloudFormation](#).

Il seguente modello CloudFormation YAML crea un gruppo di report che non esporta file di risultati di test non elaborati.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
```

```
ExportConfigType: NO_EXPORT
```

Il seguente modello CloudFormation YAML crea un gruppo di report che esporta file di risultati di test non elaborati in un bucket Amazon S3.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: S3
        S3Destination:
          Bucket: amzn-s3-demo-bucket
          Path: path-to-folder-for-exported-files
          Packaging: ZIP
          EncryptionKey: my-KMS-encryption-key
          EncryptionDisabled: false
```

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni di caricamento nel bucket S3.

Denominazione dei gruppi di report

Quando si utilizza la console AWS CLI o la AWS CodeBuild console per creare un gruppo di report, si specifica un nome per il gruppo di report. Se per la creazione ci si avvale del buildspec, il nome del nuovo gruppo presenta il formato *project-name-report-group-name-specified-in-buildspec*. Tutti i report creati eseguendo compilazioni dello stesso progetto appartengono al gruppo di report con il nuovo nome.

Se non desideri CodeBuild creare un nuovo gruppo di report, specifica l'ARN del gruppo di report nel file buildspec di un progetto di build. È possibile specificare l'ARN di uno stesso gruppo di report in più progetti di compilazione. Il gruppo di report, in tal caso, conterrà i report di test creati da ciascun progetto di compilazione a esecuzione avvenuta.

Ad esempio, se si crea un gruppo di report con il nome `my-report-group` e poi si utilizza tale nome in due diversi progetti di compilazione denominati `my-project-1` e `my-project-2` e, infine, si imposta una compilazione per entrambi i progetti, vengono creati due nuovi gruppi di report. Il risultato consiste in tre gruppi di report con i seguenti nomi:

- `my-report-group`: non annovera report di test.
- `my-project-1-my-report-group`: contiene i rapporti con i risultati dei test eseguiti dal progetto di compilazione denominato `my-project-1`.
- `my-project-2-my-report-group`: contiene i rapporti con i risultati dei test eseguiti dal progetto di compilazione denominato `my-project-2`.

Se si utilizza l'ARN del gruppo di report denominato `my-report-group` in entrambi i progetti e poi si eseguono le compilazioni di ciascun progetto, si dispone ancora di un unico gruppo di report (`my-report-group`). Tale gruppo di report contiene i rapporti con i risultati dei test eseguiti da entrambi i progetti di compilazione.

Se si sceglie un nome di gruppo di report che non appartiene a un gruppo presente nel proprio account AWS e lo si specifica in un file `buildspec`, eseguendo una compilazione del progetto correlato viene creato un nuovo gruppo di report. Il formato del nome del nuovo gruppo di report è *`project-name-new-group-name`*. Ad esempio, se nel tuo AWS account non è presente un gruppo di report con il nome `new-report-group` e specificali in un progetto di compilazione chiamato `test-project`, un'esecuzione di build crea un nuovo gruppo di report con lo stesso nome. `test-project-new-report-group`

Condividi gruppi di report

La condivisione di gruppi di report consente a più AWS account o utenti di visualizzare un gruppo di report, i report non scaduti e i risultati dei test dei relativi report. In questo modello, l'account titolare (proprietario) condivide il proprio gruppo di report con altri account (consumatori). Un consumatore non può modificare il gruppo di report. Un rapporto scade 30 giorni dopo la creazione.

Argomenti

- [Condividi un gruppo di report](#)
- [Servizi correlati](#)
- [Accedi ai gruppi di report condivisi con te](#)
- [Annullare la condivisione di un gruppo di report condiviso](#)

- [Identifica un gruppo di report condiviso](#)
- [Autorizzazioni per il gruppo di report condiviso](#)

Condividi un gruppo di report

Quando si condivide un gruppo di report, al consumatore viene concesso l'accesso in sola lettura al gruppo e ai relativi report. Il consumatore può utilizzare il AWS CLI per visualizzare il gruppo di report, i relativi report e i risultati del test case per ogni rapporto. Il consumatore non può:

- Visualizza un gruppo di report condiviso o i relativi report nella CodeBuild console.
- Modificare un gruppo di report condiviso.
- Utilizzare l'ARN del gruppo di report condiviso in un progetto, per eseguire un report. Una compilazione di progetto che specifica un gruppo di report condiviso ha esito negativo.

È possibile utilizzare la CodeBuild console per aggiungere un gruppo di report a una condivisione di risorse esistente. La [console AWS RAM](#), invece, permette di creare appositamente una nuova condivisione di risorse per il gruppo di report.

Per condividere un gruppo di report con unità organizzative o un'intera organizzazione, è necessario abilitare la condivisione con AWS Organizations. Per ulteriori informazioni, consulta [Abilitare la condivisione con AWS Organizations](#) nella Guida per l'utente di AWS RAM .

Puoi utilizzare la CodeBuild console, la AWS RAM console o AWS CLI per condividere i gruppi di report di cui sei proprietario.

Prerequisito

Per condividere un gruppo di report, il tuo AWS account deve possederlo. Un consumatore non può condividere il gruppo di report a cui ha avuto accesso.

Per condividere un gruppo di report di tua proprietà (CodeBuild console)

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Report groups (Gruppi di report).
3. Selezionare il progetto da condividere, poi fare clic su Share (Condividi). Per ulteriori informazioni, consulta l'argomento relativo alla [creazione di una condivisione di risorse](#) nella Guida per l'utente di AWS RAM .

Per condividere i gruppi di report di cui sei proprietario (console)AWS RAM

Consulta [Creating a resource share](#) in Guida per l'utente di AWS RAM .

Per condividere i gruppi di report di cui sei proprietario (AWS RAM comando)

Utilizza il comando [create-resource-share](#).

Per condividere un gruppo di report di tua proprietà (CodeBuild comando)

Utilizza il comando [put-resource-policy](#):

1. Creare un file denominato `policy.json` e copiarvi quanto segue.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"],
    "Resource": "arn:aws:iam::*:role/Service*"
  ]
}
```

2. Aggiornare `policy.json` con l'ARN del gruppo di report e gli identificatori da coinvolgere nella condivisione. L'esempio seguente concede l'accesso in sola lettura al gruppo di report con l'ARN ad Alice e all'utente root `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` per l'account identificato da 123456789012. AWS

JSON

```
{
  "Version": "2012-10-17",
```

```
"Statement": [{
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::123456789012:user/Alice",
      "123456789012"
    ]
  },
  "Action": [
    "codebuild:BatchGetReportGroups",
    "codebuild:BatchGetReports",
    "codebuild:ListReportsForReportGroup",
    "codebuild:DescribeTestCases"],
  "Resource": "arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group"
}]
}
```

3. Eseguire il seguente comando seguente.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://policy.json
```

Servizi correlati

La condivisione dei gruppi di report si integra con AWS Resource Access Manager (AWS RAM), un servizio che consente di condividere le risorse con qualsiasi account o tramite AWS Organizations. Con AWS RAM, condividi le risorse di tua proprietà creando una condivisione di risorse che specifica le risorse e i consumatori con cui condividerle. I consumatori possono appartenere a singoli AWS account AWS Organizations, a unità organizzative o a AWS Organizations un'intera organizzazione.

Per ulteriori informazioni, consulta la Guida per l'utente [AWS RAM](#).

Accedi ai gruppi di report condivisi con te

Un consumatore può accedere a un gruppo di report condiviso solo se il suo ruolo IAM dispone dell'autorizzazione BatchGetReportGroups. Puoi allegare la seguente policy al loro ruolo IAM:

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

```
"Resource": [
  "*"
],
"Action": [
  "codebuild:BatchGetReportGroups"
]
}
```

Per ulteriori informazioni, consulta [Utilizzo di politiche basate sull'identità per AWS CodeBuild](#).

Annullare la condivisione di un gruppo di report condiviso

Annullandone la condivisione, a un gruppo di report con i relativi report e risultati dei casi di test può accedere solo il proprietario. Se annulli la condivisione di un gruppo di report, qualsiasi AWS account o utente con cui lo hai condiviso in precedenza non può accedere al gruppo di report, ai relativi report o ai risultati dei test case contenuti nei report.

Per annullare, basta rimuovere il gruppo di report di proprietà dalla relativa condivisione di risorse. Puoi usare la AWS RAM console o AWS CLI farlo.

Per annullare la condivisione di un gruppo di report condiviso di tua proprietà (AWS RAM console)

Consulta [Aggiornamento di una condivisione di risorse](#) nella Guida per l'utente di AWS RAM .

Per annullare la condivisione di un gruppo di report condiviso di cui sei proprietario (comando)AWS RAM

Utilizza il comando [disassociate-resource-share](#).

Per annullare la condivisione di un gruppo di report di cui sei proprietario CodeBuild (comando)

Esegui il [delete-resource-policy](#) comando e specifica l'ARN del gruppo di report che desideri annullare la condivisione:

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Identifica un gruppo di report condiviso

Proprietari e consumatori possono AWS CLI utilizzarlo per identificare gruppi di report condivisi.

I comandi seguenti consentono di identificare e ottenere informazioni su un gruppo condiviso e i relativi report:

- Per vedere i gruppi ARNs di report condivisi con te, esegui [list-shared-report-groups](#):

```
aws codebuild list-shared-report-groups
```

- Per visualizzare i report in un gruppo ARNs di report, esegui [list-reports-for-report-group](#) utilizzando il gruppo di report ARN:

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Per visualizzare le informazioni relative ai casi di test in un report, occorre eseguire [describe-test-cases](#) utilizzando l'ARN del report:

```
aws codebuild describe-test-cases --report-arn report-arn
```

L'output sarà simile al seguente:

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

Autorizzazioni per il gruppo di report condiviso

Autorizzazioni per i proprietari

Un proprietario può modificare il gruppo di report e specificarlo in un progetto per l'esecuzione dei report.

Autorizzazioni per gli utenti

Un consumatore può visualizzare il gruppo con i relativi report e risultati dei casi di test. Un consumatore non può modificare il gruppo di report (o i relativi report) né utilizzarlo per creare altri report.

Specifica dei file di test

I file dei risultati dei test e la relativa posizione per ogni gruppo di report vanno specificati nella sezione `reports` del file `buildspec` afferente al progetto di compilazione. Per ulteriori informazioni, consulta [Reports syntax in the buildspec file](#).

Di seguito è riportata una sezione `reports` di esempio che specifica due gruppi di report per un progetto di compilazione. Uno è specificato con il suo ARN, l'altro con un nome. La sezione `files` specifica i file contenenti i risultati dei casi di test. La sezione facoltativa `base-directory` specifica la directory in cui si trovano i file dei casi di test. La facoltativa `discard-paths` sezione specifica se i percorsi dei file dei risultati dei test caricati su un bucket Amazon S3 vengono eliminati.

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  #surefire junit reports
  files:
    - '**/*'
  base-directory: 'surefire/target/surefire-reports'
  discard-paths: false

sampleReportGroup: #Cucumber reports from json plugin
  files:
    - 'cucumber-json/target/cucumber-json-report.json'
  file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Specifica dei comandi di test

È possibile specificare i comandi che eseguono i casi di test nella sezione `commands` del proprio file `buildspec`. Questi comandi eseguono i casi di test previsti per i gruppi di report nella sezione `reports` del file `buildspec`. Di seguito è riportata una sezione `commands` esemplificativa che include i comandi per l'esecuzione dei test nei file di test:

```
commands:
  - echo Running tests for surefire junit
  - mvn test -f surefire/pom.xml -fn
  - echo
  - echo Running tests for cucumber with json plugin
  - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
    cucumber-json/pom.xml -fn
```

Per ulteriori informazioni, consulta [Sintassi buildspec](#).

Aggiungi tag a un gruppo di report in AWS CodeBuild

Un tag è un'etichetta di attributo personalizzata che l'utente o AWS assegna a una AWS risorsa. Ogni AWS tag è composto da due parti:

- Una chiave di tag (ad esempio, `CostCenter`, `Environment`, `Project` o `Secret`). Le chiavi dei tag distinguono tra maiuscole e minuscole
- Un campo facoltativo noto come valore del tag (ad esempio, `111122223333`, `Production` o un nome di team). Non specificare il valore del tag equivale a utilizzare una stringa vuota. Come le chiavi tag, i valori dei tag fanno distinzione tra maiuscole e minuscole.

Tutti questi sono noti come coppie chiave-valore. Per conoscere i possibili limiti sul numero di tag in un gruppo di report e le limitazioni di chiavi e valori di tag, consulta [Tag](#).

I tag ti aiutano a identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a un gruppo di CodeBuild report che assegni a un bucket Amazon S3. Per ulteriori informazioni sull'utilizzo dei tag, consulta il whitepaper [Tagging best practices](#) (Best practice relative al tagging).

In CodeBuild, le risorse principali sono il gruppo di report e il progetto. È possibile utilizzare la CodeBuild console AWS CLI CodeBuild APIs, o AWS SDKs per aggiungere, gestire e rimuovere

tag per un gruppo di report. Oltre a identificare, organizzare e tracciare il gruppo di report con i tag, puoi utilizzare i tag nelle policy IAM per controllare chi può visualizzare e interagire con il tuo gruppo di report. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Argomenti

- [Aggiungi tag a un gruppo di report](#)
- [Visualizzazione dei tag per un gruppo di report](#)
- [Modifica tag per un gruppo di report](#)
- [Rimuovere i tag da un gruppo di report](#)

Aggiungi tag a un gruppo di report

L'aggiunta di tag a un gruppo di report può aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. In primo luogo, è possibile aggiungere uno o più tag (coppie chiave-valore) a un gruppo di report. Non dimenticare che esistono limiti sul numero di tag che è possibile applicare a un gruppo di report. Esistono limitazioni per i caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Tag](#). Dopo aver creato i tag, puoi creare policy IAM per gestire l'accesso al gruppo di report in base a questi tag. Puoi utilizzare la CodeBuild console o aggiungere tag AWS CLI a un gruppo di report.

Important

L'aggiunta di tag a un gruppo di report può influire sull'accesso a tale gruppo di report. Prima di aggiungere un tag a un gruppo di report, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare i tag per controllare l'accesso a risorse come i gruppi di report. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Per ulteriori informazioni sull'aggiunta di tag a un gruppo di report al momento della creazione, consulta [Creazione di un gruppo di report \(console\)](#).

Argomenti

- [Aggiungere un tag a un gruppo di report \(console\)](#)
- [Aggiungere un tag a un gruppo di report \(AWS CLI\)](#)

Aggiungere un tag a un gruppo di report (console)

Puoi utilizzare la CodeBuild console per aggiungere uno o più tag a un gruppo di CodeBuild report.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Gruppi di report, scegliere il nome del gruppo di report a cui si desidera aggiungere tag.
3. Nel pannello di navigazione scegli Impostazioni.
4. Se non sono stati aggiunti tag al repository, scegliere Aggiungi tag. È inoltre possibile scegliere Modifica e quindi Aggiungi tag.
5. Per Key (Chiave), inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).
6. (Facoltativo) Per aggiungere un altro tag, scegliere Add tag (Aggiungi tag) .
7. Una volta completata l'aggiunta di tag, scegliere Submit (Inoltra).

Aggiungere un tag a un gruppo di report (AWS CLI)

Per aggiungere un tag a un gruppo di report quando lo si crea, vedere [Creazione di un gruppo di report \(CLI\)](#). In `CreateReportGroup.json`, aggiungi i tuoi tag.

Per aggiungere tag a un gruppo di report esistente, vedere [Aggiornamento di un gruppo di report \(CLI\)](#) e aggiungere i tag in `UpdateReportGroupInput.json`.

In queste fasi, si assume che sia già installata una versione recente della AWS CLI o che sia aggiornata alla versione corrente. Per ulteriori informazioni, consultare [Installing the AWS Command Line Interface](#).

Visualizzazione dei tag per un gruppo di report

I tag possono aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. Per ulteriori informazioni sull'utilizzo dei tag, consulta il whitepaper [Tagging best practices](#) (Best practice relative al tagging). Per esempi di policy di accesso basate su tag, consulta [Deny or allow actions on report groups based on resource tags](#).

Visualizzazione dei tag per un gruppo di report (console)

Puoi utilizzare la CodeBuild console per visualizzare i tag associati a un gruppo di CodeBuild report.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.

2. In Gruppi di report, scegliere il nome del gruppo di report in cui si desidera visualizzare i tag.
3. Nel pannello di navigazione scegli Impostazioni.

Visualizzazione dei tag per un gruppo di report (AWS CLI)

Segui questi passaggi per utilizzare AWS CLI per visualizzare i AWS tag per un gruppo di report. Se non sono stati aggiunti tag, l'elenco restituito è vuoto.

1. Usa la console o il AWS CLI per individuare l'ARN del tuo gruppo di report. Prendere nota del valore.

AWS CLI

Eseguire il seguente comando seguente.

```
aws list-report-groups
```

Questo comando restituisce informazioni in formato JSON simili alle seguenti:

```
{
  "reportGroups": [
    "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
  ]
}
```

Un gruppo di report ARN termina con il relativo nome, che è possibile utilizzare per identificare l'ARN per il gruppo di report.

Console

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
 2. In Gruppi di report, scegliere il nome del gruppo di report con i tag che si desidera visualizzare.
 3. In Configurazione individuare l'ARN del gruppo di report.
2. Eseguire il seguente comando seguente. Utilizzate l'ARN di cui hai preso nota per il parametro `--report-group-arns`.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

In caso di esito positivo, questo comando restituisce informazioni in formato JSON che contengono una sezione tags simile alla seguente:

```
{
  ...
  "tags": {
    "Status": "Secret",
    "Project": "TestBuild"
  }
  ...
}
```

Modifica tag per un gruppo di report

È possibile modificare il valore di un tag associato a un gruppo di report. Puoi anche cambiare il nome della chiave, il che equivale a rimuovere il tag corrente e aggiungerne uno diverso con il nuovo nome e lo stesso valore dell'altra chiave. Tieni presente che esistono limitazioni sui caratteri che è possibile utilizzare nei campi chiave e valore. Per ulteriori informazioni, consulta [Tag](#).

Important

La modifica dei tag per un gruppo di report può influire sull'accesso a tale gruppo di report. Prima di modificare il nome (chiave) o il valore di un tag per un gruppo di report, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come i gruppi di report. Per esempi di policy di accesso basate su tag, consulta [Deny or allow actions on report groups based on resource tags](#).

Modificare un tag per un gruppo di report (console)

Puoi utilizzare la CodeBuild console per modificare i tag associati a un gruppo di CodeBuild report.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Gruppi di report, scegliere il nome del gruppo di report in cui si desidera modificare i tag.

3. Nel pannello di navigazione scegli Impostazioni.
4. Scegli Modifica.
5. Esegui una delle seguenti operazioni:
 - Per modificare il tag, immettere un nuovo nome in Key (Chiave). La modifica del nome del tag equivale alla sostituzione del tag esistente con un nuovo tag che ha un nuovo nome chiave.
 - Per modificare il valore di un tag, immettere un nuovo valore. Se si desidera modificare il valore in un valore vuoto, eliminare il valore corrente e lasciare il campo vuoto.
6. Al termine delle modifiche dei tag, scegliere Submit (Inoltra).

Modifica tag per un gruppo di report (AWS CLI)

Per aggiungere, modificare o eliminare tag da un gruppo di report, vedere [Aggiornamento di un gruppo di report \(CLI\)](#). Aggiorna i tag in `UpdateReportGroupInput.json`.

Rimuovere i tag da un gruppo di report

È possibile rimuovere uno o più tag associati a un gruppo di report. La rimozione di un tag non elimina il tag dalle altre AWS risorse ad esso associate.

Important

La rimozione di tag per un gruppo di report può influire sull'accesso a tale gruppo di report. Prima di rimuovere un tag da un gruppo di report, assicurati di esaminare eventuali policy IAM che potrebbero utilizzare la chiave o il valore di un tag per controllare l'accesso a risorse come i gruppi di report. Per esempi di policy di accesso basate su tag, consulta [Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse](#).

Rimuovere un tag da un gruppo di report (console)

Puoi utilizzare la CodeBuild console per rimuovere l'associazione tra un tag e un gruppo di CodeBuild report.

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. In Gruppi di report, scegliere il nome del gruppo di report in cui si desidera rimuovere i tag.
3. Nel pannello di navigazione scegli Impostazioni.
4. Scegli Modifica.

5. Cercare il tag da rimuovere, quindi scegliere Remove tag (Rimuovi tag).
6. Una volta terminata la rimozione di tag, scegliere Submit (Inoltra).

Rimuovere un tag da un gruppo di report (AWS CLI)

Segui questi passaggi per AWS CLI rimuovere un tag da un gruppo di CodeBuild report. La rimozione di un tag non lo elimina completamente, ma rimuove semplicemente l'associazione tra il tag e il gruppo di report.

Note

Se elimini un gruppo di CodeBuild report, tutte le associazioni di tag vengono rimosse dal gruppo di report eliminato. Non è necessario rimuovere i tag prima di eliminare un gruppo di report.

Per eliminare uno o più tag da un gruppo di report, vedere [Modifica tag per un gruppo di report \(AWS CLI\)](#). Aggiornare la sezione tags nei dati formattati JSON con un elenco aggiornato di tag che non contiene quelli che si desidera eliminare. Se si desidera eliminare tutti i tag, aggiornare la sezione tags per:

```
"tags: []"
```

Aggiornamento di un gruppo di report

Quando aggiorni un gruppo di report, puoi specificare se esportare i dati non elaborati dei risultati del test in file in un bucket Amazon S3. Se si opta per l'esportazione in un bucket S3, è necessario specificare quanto segue per il gruppo di report:

- L'eventuale compressione in un file ZIP dei file con i risultati dei test non elaborati.
- L'eventuale crittografia dei file con i risultati dei test non elaborati. È possibile definire la crittografia con una delle seguenti opzioni:
 - E Chiave gestita da AWS per Amazon S3.
 - Una chiave gestita dal cliente che puoi creare e configurare.

Per ulteriori informazioni, consulta [Crittografia dei dati](#).

Se utilizzi il AWS CLI per aggiornare un gruppo di report, puoi anche aggiornare o aggiungere tag. Per ulteriori informazioni, consulta [Aggiungi tag a un gruppo di report in AWS CodeBuild](#).

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni al caricamento nel bucket S3.

Argomenti

- [Aggiornamento di un gruppo di report \(console\)](#)
- [Aggiornamento di un gruppo di report \(CLI\)](#)

Aggiornamento di un gruppo di report (console)

Utilizza la seguente procedura per aggiornare un gruppo di report utilizzando Console di gestione AWS

Per aggiornare un gruppo di report

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Report groups (Gruppi di report).
3. Scegliere il gruppo di report da aggiornare.
4. Scegli Modifica.
5. Seleziona o deseleziona Backup su Amazon S3. Con questa opzione selezionata, occorre specificare le impostazioni di esportazione:
 - a. Immettere il nome del proprio bucket S3 in S3 bucket name (Nome bucket S3).
 - b. Per Path prefix (Prefisso percorso), immettere il percorso di caricamento dei risultati del test nel bucket S3.
 - c. Per comprimere i file di dati non elaborati con i risultati del test, selezionare Compress test result data in a zip file (Comprimi risultati test in file zip).
 - d. Per visualizzare le opzioni di crittografia, espandere Additional configuration (Configurazione aggiuntiva). Seleziona una delle seguenti opzioni:

- Chiave AWS gestita predefinita da utilizzare Chiave gestita da AWS per Amazon S3. Per ulteriori informazioni, consulta [Customer managed CMKs](#) in the AWS Key Management Service User Guide. Questa è l'opzione di crittografia predefinita.
- Scegli una chiave personalizzata per utilizzare una chiave gestita dal cliente che crei e configuri. In AWS KMS encryption key (Chiave di crittografia), immettere l'ARN della chiave di crittografia. Il formato è: `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Per ulteriori informazioni, consulta [Creazione di chiavi KMS](#) nella Guida per l'utente di AWS Key Management Service .
- Disable artifact encryption (Disabilita crittografia artefatti) consente di disabilitare la crittografia. Bisogna scegliere questa opzione se si desidera condividere i risultati dei test o pubblicarli su un sito Web statico. (Un sito Web dinamico può eseguire codice per decrittare i risultati dei test.)

Aggiornamento di un gruppo di report (CLI)

Utilizzare la procedura seguente per aggiornare un gruppo di report utilizzando AWS CLI.

Per aggiornare un gruppo di report

1. Crea un file denominato `UpdateReportGroupInput.json`.
2. Copiare quanto segue in `UpdateReportGroupInput.json`.

```
{
  "arn": "",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "bucket-name",
      "path": "path",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "your-key"
    }
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```

```
]
}
```

3. Immettere l'ARN del gruppo di report nella riga `arn` (ad esempio, `"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1")`)
4. Aggiornare `UpdateReportGroupInput.json` con gli aggiornamenti che si desidera applicare al gruppo di report.
 - Se si desidera aggiornare il gruppo di report per esportare i file dei risultati del test grezzi in un bucket S3, aggiornare la sezione `exportConfig`. Sostituiscilo `bucket-name` con il nome del bucket S3 e `path` con il percorso nel bucket S3 in cui desideri esportare i file. Se si desidera comprimere i file esportati, in `packaging` specificare `ZIP`. In caso contrario, specificare `NONE`. Utilizzare `encryptionDisabled` per stabilire se crittografare i file esportati. Se crittografi i file esportati, inserisci la chiave gestita dal cliente.
 - Se si desidera aggiornare il gruppo di report in modo che non esporti i file dei risultati dei test non elaborati in un bucket S3, aggiornare la sezione `exportConfig` con il seguente JSON:

```
{
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

- Se si desidera aggiornare i tag del gruppo di report, aggiornare la sezione `tags`. È possibile modificare, aggiungere o rimuovere tag. Se si desidera rimuovere tutti i tag, aggiornarlo con il seguente JSON:

```
"tags": []
```

5. Esegui il comando seguente:

```
aws codebuild update-report-group \
--cli-input-json file:///UpdateReportGroupInput.json
```

Framework di test

Gli argomenti di questa sezione mostrano come impostare la reportistica dei test per vari framework di test. AWS CodeBuild

Argomenti

- [Impostare i report dei test con Jasmine](#)
- [Impostare i report dei test con Jest](#)
- [Impostare i report dei test con pytest](#)
- [Imposta la reportistica dei test con RSpec](#)

Impostare i report dei test con Jasmine

La procedura seguente mostra come configurare la reportistica dei test AWS CodeBuild con il framework di test [JasmineBDD](#).

La procedura richiede i seguenti prerequisiti:

- Hai un progetto esistente. CodeBuild
- Il tuo progetto è un progetto Node.js impostato per utilizzare il framework di test Jasmine.

Aggiungi il [jasmine-reporters](#) pacchetto alla sezione `devDependencies` del file `package.json` del tuo progetto. Questo pacchetto contiene una raccolta di classi JavaScript reporter che possono essere utilizzate con Jasmine.

```
npm install --save-dev jasmine-reporters
```

Se non è già presente, aggiungi lo script `test` al file `package.json` del tuo progetto. Lo script `test` assicura che Jasmine venga chiamato quando `npm test` viene eseguito.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild supporta i seguenti reporter di test di Jasmine:

JUnitXmlReporter

Utilizzato per generare report nel formato JUnitXml.

NUnitXmlReporter

Utilizzato per generare report nel formato NUnitXml.

Un progetto Node.js con Jasmine avrà, per impostazione predefinita, una sottodirectory `spec`, che contiene gli script di configurazione e test Jasmine.

Per configurare Jasmine e generare report nel formato JUnitXML, creare un'istanza del reporter `JUnitXmlReporter` aggiungendo il seguente codice ai test.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(junitReporter);
```

Per configurare Jasmine e generare report nel formato NUnitXML, creare un'istanza del reporter `NUnitXmlReporter` aggiungendo il seguente codice ai test.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(nunitReporter)
```

I report dei test vengono esportati nel file specificato da/. *<test report directory> <report filename>*

Nel file `buildspec.yml`, aggiungere/aggiornare le sezioni seguenti.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jasmine_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Se si utilizza il formato del report NunitXml, modificare il valore `file-format` nel modo seguente.

```
file-format: NUNITXML
```

Impostare i report dei test con Jest

La seguente procedura mostra come configurare la reportistica dei test AWS CodeBuild con il framework di test [Jest](#).

La procedura richiede i seguenti prerequisiti:

- Hai un progetto esistente CodeBuild .
- Il progetto è un progetto Node.js impostato per utilizzare il framework di test Jest.

Aggiungi il [jest-junit](#) pacchetto alla `devDependencies` sezione del `package.json` file del tuo progetto. CodeBuild utilizza questo pacchetto per generare report nel JunitXml formato.

```
npm install --save-dev jest-junit
```

Se non è già presente, aggiungi lo script `test` al file `package.json` del tuo progetto. Lo `test` script assicura che Jest venga chiamato quando `npm test` viene eseguito.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Configurare Jest per utilizzare il reporter JunitXml aggiungendo quanto segue al file di configurazione Jest. Se il progetto non dispone di un file di configurazione Jest, creare un file denominato `jest.config.js` nella radice del progetto e aggiungere quanto segue. I report dei test vengono esportati nel file specificato da `<test report directory>/. <report filename>`

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
  ]
};
```

Nel file `buildspec.yml`, aggiungere/aggiornare le sezioni seguenti.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jest_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Impostare i report dei test con pytest

La seguente procedura mostra come configurare la reportistica dei test AWS CodeBuild con il framework di [test pytest](#).

La procedura richiede i seguenti prerequisiti:

- Hai un progetto esistente CodeBuild .
- Il tuo progetto è un progetto Python che è impostato per utilizzare il framework di test pytest.

Aggiungere la seguente voce alla fase build o post_build del file buildspec.yml. Questo codice rileva automaticamente i test nella directory corrente ed esporta i report dei test nel file specificato da *<test report directory>/<report filename>*. Il report utilizza il formato JunitXml.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

Nel file buildspec.yml, aggiungere/aggiornare le sezioni seguenti.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Imposta la reportistica dei test con RSpec

La procedura seguente mostra come configurare i report sui test AWS CodeBuild con il [framework di RSpec test](#).

La procedura richiede i seguenti prerequisiti:

- Hai un CodeBuild progetto esistente.
- Il tuo progetto è un progetto Ruby configurato per utilizzare il framework di RSpec test.

Aggiungi/aggiorna quanto segue nel tuo file `buildspec.yml`. Questo codice esegue i test nella `<test source directory>` directory ed esporta i report dei test nel file specificato da `<test report directory>/<report filename>`. Il report utilizza il formato JUnitXML.

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  pre_build:
    commands:
      - gem install rspec
      - gem install rspec_junit_formatter
  build:
    commands:
      - rspec <test source directory>/* --format RspecJUnitFormatter --out <test report
        directory>/<report filename>
reports:
  rspec_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Visualizzazione dei report di test

È possibile reperire informazioni dettagliate su un report di test e quindi sui suoi casi, quali i tassi di riuscita ed errore e il tempo di esecuzione. Puoi visualizzare i report di test raggruppati per build,

esecuzione della build, gruppo di report o account. AWS Scegliendo un report di test nella console, se ne visualizzeranno i dettagli con i risultati dei casi di test correlati.

È possibile visualizzare solo i report di test non scaduti. I report dei test scadono 30 giorni dopo la creazione. Non è possibile visualizzare un report scaduto in CodeBuild.

Argomenti

- [Visualizzazione dei report di test in una compilazione](#)
- [Visualizzazione dei report di test in un gruppo di report](#)
- [Visualizza i report dei test nel tuo account AWS](#)

Visualizzazione dei report di test in una compilazione

Come visualizzare i report di test relativi a una compilazione

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Individuare la compilazione da visualizzare. Se si conosce il progetto esecutore della compilazione che ha creato il report di test:
 1. Nel riquadro di navigazione, selezionare Build projects (Progetti di compilazione) e poi scegliere il progetto con la compilazione che ha eseguito il report di test da visualizzare.
 2. Selezionare Build history (Cronologia delle compilazioni), poi scegliere la compilazione eseguita per creare i report da visualizzare.

Si può anche individuare la compilazione nella cronologia dedicata afferente al proprio account AWS :

1. Nel riquadro di navigazione, selezionare Build history (Cronologia delle compilazioni) e poi scegliere la compilazione che ha creato i report di test da visualizzare.
3. Nella pagina della compilazione, selezionare innanzitutto Reports (Report) e poi scegliere un report di test per visualizzarne i dettagli.

Visualizzazione dei report di test in un gruppo di report

Come visualizzare i report di test in un gruppo di report

1. [Apri la AWS CodeBuild console su codebuild/home. https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Nel riquadro di navigazione, scegliere Report groups (Gruppi di report).
3. Selezionare il gruppo di report contenente i report di test da visualizzare.
4. Scegliere un report di test per visualizzarne i dettagli.

Visualizza i report dei test nel tuo account AWS

Per visualizzare i rapporti sui test nel tuo AWS account

1. Apri la AWS CodeBuild console su <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Nel riquadro di navigazione, scegliere Report history (Cronologia report).
3. Scegliere un report di test per visualizzarne i dettagli.

Autorizzazioni per il rapporto di prova

In questa sezione sono riportate informazioni fondamentali sulle autorizzazioni correlate ai report di test.

Argomenti

- [Ruolo IAM per i report di test](#)
- [Autorizzazioni per le operazioni correlate ai report di test](#)
- [Esempi di autorizzazioni per i report di test](#)

Ruolo IAM per i report di test

Per eseguire un report di test e per aggiornare un progetto in modo da includere i report di test, il ruolo IAM necessita delle autorizzazioni seguenti. Queste autorizzazioni sono incluse nelle politiche AWS gestite predefinite. Per includere i report di test in un progetto di compilazione esistente, è necessario aggiungere queste autorizzazioni manualmente.

- `CreateReportGroup`

- CreateReport
- UpdateReport
- BatchPutTestCases

Per eseguire un rapporto sulla copertura del codice, il tuo ruolo IAM deve includere anche l'autorizzazione BatchPutCodeCoverages.

Note

BatchPutTestCases, CreateReportUpdateReport, e non BatchPutCodeCoverages sono autorizzazioni pubbliche. Non è possibile chiamare un AWS CLI comando o un metodo SDK corrispondente per queste autorizzazioni.

Per assicurarti di disporre di queste autorizzazioni, puoi allegare la seguente policy al tuo ruolo IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Si consiglia di limitare questa policy ai soli gruppi di report da utilizzare. Quanto segue limita le autorizzazioni solo ai gruppi di report i cui due sono inclusi ARNs nella policy:

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
  ]
}
```



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

Quanto segue limita le autorizzazioni ai soli gruppi di report creati mediante l'esecuzione di compilazioni di un progetto denominato `my-project`:

```
{  
    "Effect": "Allow",  
    "Resource": [  
        "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"  
    ],  
    "Action": [  
        "codebuild:CreateReportGroup",  
        "codebuild:CreateReport",  
        "codebuild:UpdateReport",  
        "codebuild:BatchPutTestCases",  
        "codebuild:BatchPutCodeCoverages"  
    ]  
}
```

Note

Il ruolo CodeBuild di servizio specificato nel progetto viene utilizzato per le autorizzazioni di caricamento nel bucket S3.

Autorizzazioni per le operazioni correlate ai report di test

Puoi specificare le autorizzazioni per le seguenti operazioni API di test reporting: CodeBuild

- `BatchGetReportGroups`
- `BatchGetReports`
- `CreateReportGroup`

- DeleteReportGroup
- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

Per ulteriori informazioni, consulta [AWS CodeBuild riferimento alle autorizzazioni](#).

Esempi di autorizzazioni per i report di test

Per informazioni sulle policy esemplificative correlate ai report di test, consulta:

- [Concessione a un utente di modificare un gruppo di report](#)
- [Concessione a un utente di creare un gruppo di report](#)
- [Concessione a un utente di eliminare un report](#)
- [Concessione a un utente di eliminare un gruppo di report](#)
- [Concessione a un utente di ottenere informazioni sui gruppi di report](#)
- [Concessione a un utente di ottenere informazioni sui report](#)
- [Concessione a un utente di ottenere un elenco dei gruppi di report](#)
- [Concessione a un utente di ottenere un elenco dei report](#)
- [Concessione a un utente di ottenere un elenco dei report riferiti a un gruppo di report](#)
- [Concessione a un utente di ottenere un elenco dei casi di test riferiti a un report](#)

Stati del rapporto di test

Un report di test può trovarsi in uno dei seguenti stati:

- GENERATING: l'esecuzione dei casi di test è ancora in corso.
- DELETING: il report di test viene eliminato. L'eliminazione di un report di test comporta la rimozione di tutti i suoi casi di test. I file di dati grezzi risultanti dai test, se esportati in un bucket S3, non vengono eliminati.

- **INCOMPLETE:** il report di test non è stato completato. Questo stato potrebbe essere restituito per uno dei seguenti motivi:
 - Un problema con la configurazione del gruppo di report che specifica i casi di test di questo report. Ad esempio, il percorso dei casi di test afferenti al gruppo di report nel file buildspec potrebbe non essere corretto.
 - L'utente IAM che ha eseguito la compilazione non dispone delle autorizzazioni necessarie per effettuare i test. Per ulteriori informazioni, consulta [Autorizzazioni per il rapporto di prova](#).
 - La compilazione non è stata completata a causa di un errore non correlato ai test.
- **SUCCEEDED:** tutti i casi di test sono riusciti.
- **FAILED:** alcuni casi di test non hanno avuto esito positivo.

Ogni caso di test restituisce uno stato. Un caso di test può trovarsi in uno dei seguenti stati:

- **SUCCEEDED:** il caso di test è riuscito.
- **FAILED:** il caso di test non è riuscito.
- **ERROR:** il caso di test ha generato un errore imprevisto.
- **SKIPPED:** il caso di test non è stato eseguito.
- **UNKNOWN:** il caso di test ha restituito uno stato diverso da SUCCEEDED, FAILED, ERROR e SKIPPED.

Un report di test può annoverare al massimo 500 risultati di casi di test. Se vengono eseguiti più di 500 casi di test, CodeBuild assegna la priorità ai test in base allo stato FAILED e tronca i risultati dei test case.

Utilizzo AWS CodeBuild con Amazon Virtual Private Cloud

In genere, AWS CodeBuild non è possibile accedere alle risorse in un VPC. Per abilitare l'accesso, è necessario fornire ulteriori informazioni di configurazione specifiche per VPC CodeBuild nella configurazione del progetto. Ciò include l'ID VPC, la sottorete VPC IDs e il gruppo di sicurezza VPC. IDs Le compilazioni attivate da VPC possono accedere alle risorse all'interno del tuo VPC. [Per ulteriori informazioni sulla configurazione di un VPC in Amazon VPC, consulta la Amazon VPC User Guide.](#)

Argomenti

- [Casi d'uso](#)
- [Le migliori pratiche per VPCs](#)
- [Limitazioni di VPCs](#)
- [Consenti l'accesso ad Amazon VPC nei tuoi progetti CodeBuild](#)
- [Risolvi i problemi relativi alla configurazione del VPC](#)
- [Utilizzo degli endpoint VPC](#)
- [Utilizzo AWS CodeBuild con un server proxy gestito](#)
- [Utilizzo AWS CodeBuild con un server proxy](#)
- [CloudFormation Modello VPC](#)

Casi d'uso

La connettività VPC AWS CodeBuild delle build consente di:

- Esegui i test di integrazione della tua build rispetto ai dati in un database Amazon RDS isolato su una sottorete privata.
- Interroga i dati in un ElastiCache cluster Amazon direttamente dai test.
- Interagisci con servizi Web interni ospitati su Amazon EC2, Amazon ECS o servizi che utilizzano Elastic Load Balancing interno.
- Recuperare dipendenze da repository di artefatti interni e autogestiti, come PyPI per Python, Maven per Java e npm per Node.js.
- Accedi agli oggetti in un bucket S3 configurato per consentire l'accesso solo tramite un endpoint Amazon VPC.

- Eseguire una query dei servizi Web esterni che richiedono indirizzi IP fissi attraverso l'indirizzo IP elastico del gateway NAT o dell'istanza NAT associata alla sottorete.

Le build possono accedere a tutte le risorse ospitate sul tuo VPC.

Le migliori pratiche per VPCs

Usa questa lista di controllo quando configuri un VPC con cui lavorare. CodeBuild

- Configura il tuo VPC con sottoreti pubbliche e private e un gateway NAT. Il gateway NAT deve risiedere in una sottorete pubblica. Per ulteriori informazioni, consulta [VPC con sottoreti pubbliche e private \(NAT\)](#) in Guida per l'utente di Amazon VPC.

Important

È necessario un gateway NAT o un'istanza NAT da utilizzare CodeBuild con il VPC in modo che CodeBuild possa raggiungere gli endpoint pubblici (ad esempio, per eseguire i comandi CLI durante l'esecuzione delle build). Non è possibile utilizzare il gateway Internet anziché un gateway NAT o un'istanza NAT perché CodeBuild non supporta l'assegnazione di indirizzi IP elastici alle interfacce di rete che crea e l'assegnazione automatica di un indirizzo IP pubblico non è supportata da Amazon EC2 per le interfacce di rete create al di fuori del lancio delle istanze Amazon EC2.

- Includere più zone di disponibilità al VPC.
- Assicurati che ai tuoi gruppi di sicurezza non sia consentito il traffico in entrata (in ingresso) verso le tue build. CodeBuild non ha requisiti specifici per il traffico in uscita, ma devi consentire l'accesso a tutte le risorse Internet necessarie per la tua build, ad GitHub esempio Amazon S3.

Per ulteriori informazioni, consulta [Regole del gruppo di sicurezza](#) nella Guida per l'utente di Amazon VPC.

- Impostare sottoreti separate per le build.
- Quando configuri i tuoi CodeBuild progetti per accedere al tuo VPC, scegli solo sottoreti private.

[Per ulteriori informazioni sulla configurazione di un VPC in Amazon VPC, consulta la Amazon VPC User Guide.](#)

Per ulteriori informazioni sull'utilizzo della funzionalità VPC CloudFormation per configurare un CodeBuild VPC, consulta la. [CloudFormation Modello VPC](#)

Limitazioni di VPCs

- La connettività VPC di non CodeBuild è supportata per la condivisione. VPCs

Consenti l'accesso ad Amazon VPC nei tuoi progetti CodeBuild

Includi queste impostazioni nella configurazione del VPC:

- Per VPC ID, scegli l'ID VPC che utilizza. CodeBuild
- Per le sottoreti, scegli una sottorete privata con traduzione NAT che includa o disponga di percorsi verso le risorse utilizzate da. CodeBuild
- Per i gruppi di sicurezza, scegliete i gruppi di sicurezza da CodeBuild utilizzare per consentire l'accesso alle risorse in. VPCs

Per utilizzare la console per creare un progetto di compilazione, consulta [Creare un progetto di compilazione \(console\)](#). Quando crei o modifichi il tuo CodeBuild progetto, in VPC, scegli l'ID VPC, le sottoreti e i gruppi di sicurezza.

Per utilizzare il per creare un AWS CLI progetto di compilazione, vedi. [Creazione di un progetto di compilazione \(AWS CLI\)](#) Se utilizzi AWS CLI with CodeBuild, il ruolo di servizio utilizzato da CodeBuild per interagire con i servizi per conto dell'utente IAM deve avere una policy allegata. Per informazioni, consulta [Consenti CodeBuild l'accesso ai AWS servizi necessari per creare un'interfaccia di rete VPC](#).

L'*vpcConfig* oggetto dovrebbe includere il tuo *vpcIdsecurityGroupIds*, e *subnets*.

- *vpcId*: richiesto. L'ID VPC che CodeBuild utilizza. Esegui questo comando per ottenere un elenco di tutti gli Amazon VPC IDs nella tua regione:

```
aws ec2 describe-vpcs
```

- *subnets*: richiesto. La sottorete IDs che include le risorse utilizzate da. CodeBuild Esegui questo comando per ottenere questi IDs:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

 Note

Sostituisci us-east-1 con la tua regione.

- ***securityGroupIds***: richiesto. Il gruppo di sicurezza IDs utilizzato da CodeBuild per consentire l'accesso alle risorse in VPCs. Eseguite questo comando per ottenere quanto segue IDs:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

 Note

Sostituisci us-east-1 con la tua regione.

Risolvi i problemi relativi alla configurazione del VPC

Utilizza le informazioni presenti nel messaggio di errore per identificare, diagnosticare e risolvere i problemi.

Di seguito sono riportate alcune linee guida per la risoluzione di un errore VPC CodeBuild comune: Build does not have internet connectivity. Please check subnet network configuration.

1. [Assicurarsi che il gateway Internet sia collegato al VPC.](#)
2. [Assicurarsi che la tabella di routing della sottorete pubblica punti all'Internet Gateway.](#)
3. [Assicurati che la tua rete ACLs consenta il flusso del traffico.](#)
4. [Assicurarsi che i gruppi di sicurezza consentano il traffico sul flusso.](#)
5. [Risolvere i problemi del gateway NAT.](#)
6. [Assicurarsi che la tabella di routing delle sottoreti private punti al gateway NAT.](#)
7. Assicurati che il ruolo di servizio utilizzato da CodeBuild per interagire con i servizi per conto dell'utente IAM disponga delle autorizzazioni previste da [questa policy](#). Per ulteriori informazioni, consulta [Consenti CodeBuild di interagire con altri servizi AWS](#).

Se CodeBuild mancano le autorizzazioni, potresti ricevere un errore che dice, `Unexpected EC2 error: UnauthorizedOperation`. Questo errore può verificarsi se CodeBuild non si dispone delle autorizzazioni Amazon EC2 necessarie per lavorare con un VPC.

Utilizzo degli endpoint VPC

Puoi migliorare la sicurezza delle tue build configurando l'uso di un AWS CodeBuild endpoint VPC di interfaccia. Gli endpoint di interfaccia sono alimentati da PrivateLink, una tecnologia che puoi utilizzare per accedere in modo privato ad Amazon EC2 CodeBuild e utilizzare indirizzi IP privati. PrivateLink limita tutto il traffico di rete tra le istanze gestite e Amazon EC2 alla rete Amazon. CodeBuild Le istanze gestite non hanno accesso a Internet. Inoltre, non hai bisogno di un gateway Internet, di un dispositivo NAT o di un gateway privato virtuale. Non è obbligatorio configurare PrivateLink, ma è consigliato. [Per ulteriori informazioni sugli PrivateLink endpoint VPC, consulta Cos'è? AWS PrivateLink](#).

Prima di creare endpoint VPC

Prima di configurare gli endpoint VPC per AWS CodeBuild, tieni presente le seguenti restrizioni e limitazioni.

Note

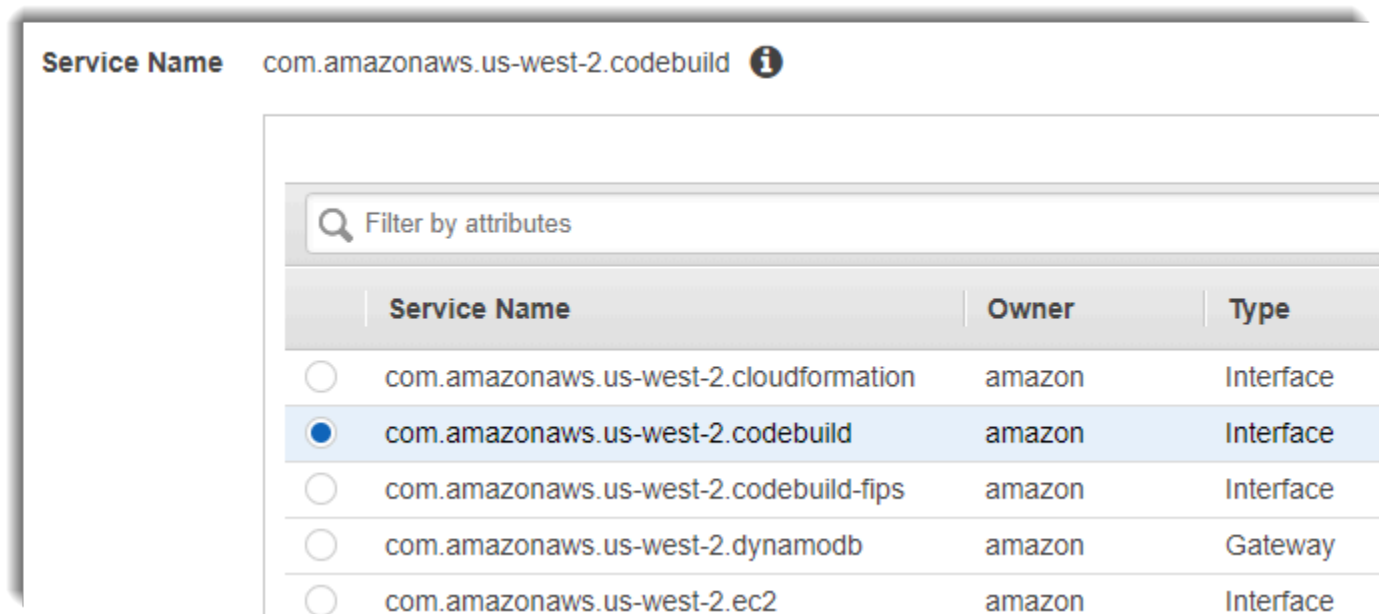
Utilizza un [gateway NAT](#) se desideri utilizzarlo CodeBuild con AWS servizi che non supportano le connessioni Amazon PrivateLink VPC.

- Gli endpoint VPC supportano il DNS fornito da Amazon solo tramite Amazon Route 53. Se si desidera utilizzare il proprio DNS, è possibile usare l'inoltro condizionale sul DNS. Per ulteriori informazioni, consulta [Set di opzioni DHCP](#) nella Guida per l'utente di Amazon VPC.
- Gli endpoint VPC attualmente non supportano le richieste tra regioni. Assicurati di creare il tuo endpoint nella stessa AWS regione di tutti i bucket S3 che memorizzano l'input e l'output della build. Puoi utilizzare la console Amazon S3 o il [get-bucket-location](#) comando per trovare la posizione del tuo bucket. Usa un endpoint Amazon S3 specifico della regione per accedere al tuo bucket (ad esempio,). `<bucket-name>.s3-us-west-2.amazonaws.com` Per ulteriori informazioni sugli endpoint specifici della regione per Amazon S3, consulta [Amazon Simple Storage Service](#) nel. Riferimenti generali di Amazon Web Services Se utilizzi il AWS CLI per

effettuare richieste ad Amazon S3, imposta la tua regione predefinita sulla stessa regione in cui è stato creato il bucket o utilizza il `--region` parametro nelle tue richieste.

Crea endpoint VPC per CodeBuild

Segui le istruzioni contenute in [Creazione di un endpoint dell'interfaccia](#) per creare l'endpoint `com.amazonaws.us-west-2.codebuild`. Questo è un endpoint VPC per AWS CodeBuild



region rappresenta l'identificatore di regione per una AWS regione supportata da CodeBuild, ad esempio `us-east-2` per la regione Stati Uniti orientali (Ohio). Per un elenco delle AWS regioni supportate, consulta la [CodeBuild Guida AWS generale](#). L'endpoint è precompilato con la regione specificata al momento dell'accesso. AWS Se modifichi la regione, l'endpoint VPC viene aggiornato di conseguenza.

Crea una policy per gli endpoint VPC per CodeBuild

Puoi creare una policy per gli endpoint Amazon VPC per la AWS CodeBuild quale puoi specificare:

- Il principale che può eseguire azioni.
- Le operazioni che possono essere eseguite.
- Le risorse su cui è possibile eseguire le operazioni.

L'esempio di policy seguente specifica che tutti i principali possono solo iniziare e visualizzare le build per il progetto `project-name`.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) nella Guida per l'utente di Amazon VPC.

Utilizzo AWS CodeBuild con un server proxy gestito

Per eseguire flotte a capacità AWS CodeBuild riservata in un server proxy gestito, è necessario configurare il server proxy in modo da consentire o negare il traffico da e verso siti esterni utilizzando le regole proxy. Tieni presente che l'esecuzione di flotte a capacità riservata in un server proxy gestito non è supportata per VPC, Windows o macOS.

Important

Esistono costi aggiuntivi in base alla durata della presenza di una configurazione proxy nel parco veicoli. Per ulteriori informazioni, consulta <https://aws.amazon.com/codebuild/pricing/>.

Argomenti

- [Configurare una configurazione proxy gestita per flotte a capacità riservata](#)
- [Gestisci un parco veicoli con capacità CodeBuild riservata](#)

Configurare una configurazione proxy gestita per flotte a capacità riservata

Per configurare un server proxy gestito per il tuo parco veicoli a capacità riservata, devi abilitare questa funzionalità quando crei il tuo parco veicoli nella console o utilizzi il AWS CLI. È necessario definire diverse proprietà:

Definire le configurazioni del proxy (opzionale)

Configurazioni proxy che applicano il controllo dell'accesso alla rete alle istanze con capacità riservata.

Comportamento predefinito

Definisce il comportamento del traffico in uscita.

Abilita

Per impostazione predefinita, consente il traffico in uscita verso tutte le destinazioni.

Rifiuta

Per impostazione predefinita, nega il traffico in uscita verso tutte le destinazioni.

Regole proxy

Specifica i domini di destinazione a cui limitare il controllo dell'accesso alla rete.

Per definire le configurazioni del proxy nella console, consulta [Crea un parco veicoli a capacità riservata](#) le istruzioni. Per definire le configurazioni del proxy utilizzando AWS CLI, puoi farlo modificando la seguente sintassi JSON e salvando i risultati:

```
"proxyConfiguration": {
  "defaultBehavior": "ALLOW_ALL" | "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN" | "IP",
      "effect": "ALLOW" | "DENY",
      "entities": [
        "destination"
      ]
    }
  ]
}
```

Il file JSON potrebbe essere simile al seguente:

```
"proxyConfiguration": {
  "defaultBehavior": "DENY_ALL",
  "orderedProxyRules": [
    {
      "type": "DOMAIN",
      "effect": "ALLOW",
      "entities": [
        "github.com"
      ]
    }
  ]
}
```

Gestisci un parco veicoli con capacità CodeBuild riservata

Quando gestisci flotte a capacità AWS CodeBuild riservata con il tuo server proxy gestito, CodeBuild imposterà automaticamente le relative variabili HTTP_PROXY e di HTTPS_PROXY ambiente con gli indirizzi proxy gestiti. Se il tuo software di dipendenza ha una propria configurazione e non rispetta le variabili di ambiente, puoi fare riferimento a questi valori e aggiornare la configurazione del software nei comandi di build per indirizzare correttamente il traffico di build attraverso il proxy gestito. Per ulteriori informazioni, consultare [Crea un progetto di compilazione in AWS CodeBuild](#) e [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).

Utilizzo AWS CodeBuild con un server proxy

È possibile utilizzarlo AWS CodeBuild con un server proxy per regolare il traffico HTTP e HTTPS da e verso Internet. Per funzionare CodeBuild con un server proxy, installi un server proxy in una sottorete pubblica e CodeBuild in una sottorete privata in un VPC.

Esistono due casi d'uso principali per l'esecuzione CodeBuild in un server proxy:

- Elimina la necessità di utilizzare un gateway NAT o un'istanza NAT nel VPC.
- Consente di specificare a quali istanze del server proxy possono accedere e URLs a quali il server proxy nega l'accesso. URLs

È possibile utilizzarlo CodeBuild con due tipi di server proxy. Per entrambi, il server proxy viene eseguito in una sottorete pubblica e CodeBuild in una sottorete privata.

- Proxy esplicito: se si utilizza un server proxy esplicito, è necessario configurare NO_PROXY e inserire le variabili di HTTPS_PROXY ambiente a CodeBuild livello di progetto. HTTP_PROXY Per ulteriori informazioni, consultare [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#) e [Crea un progetto di compilazione in AWS CodeBuild](#).
- Proxy trasparente: se utilizzi un server proxy trasparente, non è richiesta alcuna configurazione speciale.

Argomenti

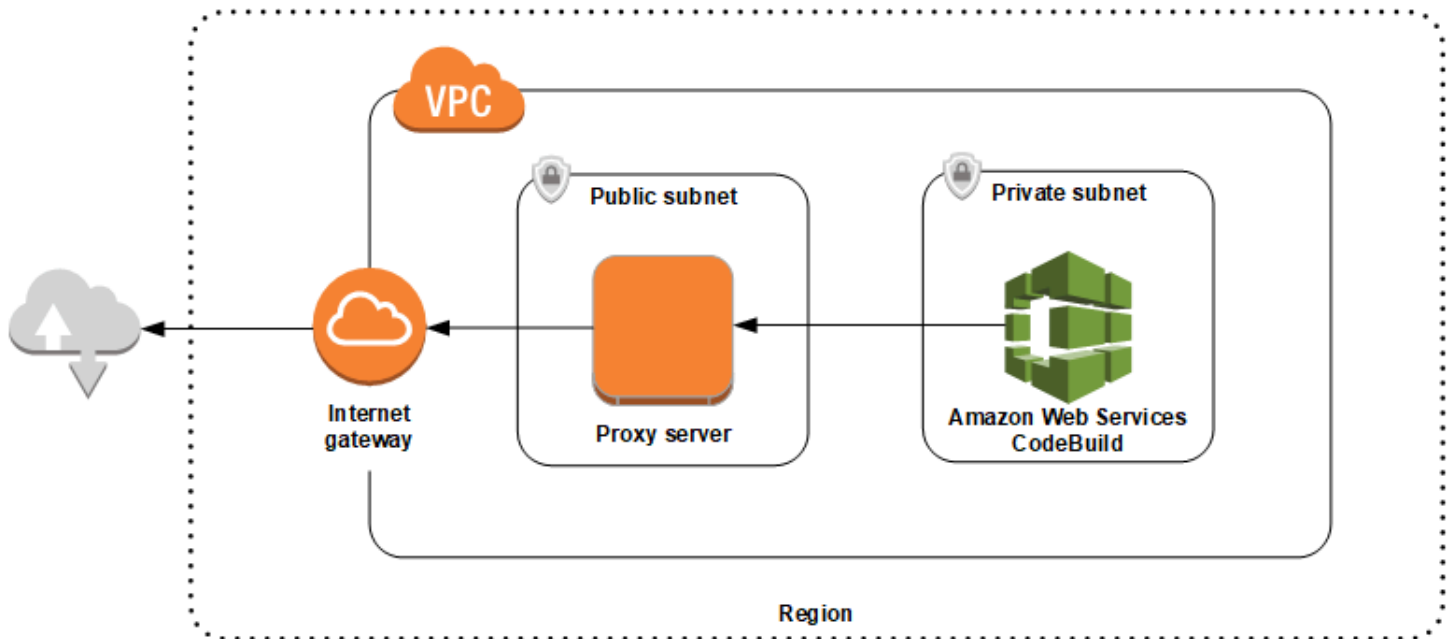
- [Configura i componenti necessari per l'esecuzione CodeBuild in un server proxy](#)
- [Esegui CodeBuild in un server proxy esplicito](#)
- [Esegui CodeBuild in un server proxy trasparente](#)
- [Esegui un programma di gestione dei pacchetti e di altri strumenti in un server proxy](#)

Configura i componenti necessari per l'esecuzione CodeBuild in un server proxy

Sono necessari questi componenti per l'esecuzione AWS CodeBuild in un server proxy trasparente o esplicito:

- Un VPC.
- Una sottorete pubblica nel VPC per il server proxy.
- Una sottorete privata nel VPC per CodeBuild.
- Un gateway Internet che consente la comunicazione tra il VPC e Internet.

Il diagramma riportato di seguito illustra l'interazione tra i componenti.



Configura un VPC, sottoreti e un gateway di rete

I seguenti passaggi sono necessari per l'esecuzione AWS CodeBuild in un server proxy trasparente o esplicito.

1. Crea un VPC. Per informazioni, consulta [Creazione di un VPC](#) nella Guida per l'utente di Amazon VPC.
2. Crea due sottoreti nel VPC. Una è una sottorete pubblica denominata `Public Subnet` in cui viene eseguito il server proxy. L'altra è una sottorete privata denominata `Private Subnet` in cui CodeBuild viene eseguito.

Per ulteriori informazioni, consulta [Creazione di una sottorete nel VPC](#).

3. Crea e collega un gateway Internet al VPC. Per ulteriori informazioni, consulta l'argomento relativo alla [creazione e all'associazione di un gateway Internet](#).
4. Aggiungi una regola alla tabella di routing predefinita che instrada il traffico in uscita dal VPC (0.0.0.0/0) al gateway Internet. Per ulteriori informazioni, consulta [Aggiunta e rimozione di route da una tabella di routing](#).
5. Aggiungi una regola al gruppo di sicurezza predefinito del VPC che consente l'ingresso del traffico SSH (TCP 22) proveniente dal VPC (0.0.0.0/0).
6. Segui le istruzioni in [Avvio di un'istanza utilizzando la procedura guidata di avvio dell'istanza](#) nella Guida per l'utente di Amazon EC2 per avviare un'istanza Amazon Linux. Durante la procedura guidata, scegli le seguenti opzioni:

- In Scegli un tipo di istanza, scegli un'Amazon Linux Amazon Machine Image (AMI).
- In Subnet (Sottorete), scegli la sottorete pubblica precedentemente creata in questo argomento. Se è stato utilizzato il nome suggerito, è Public Subnet (Sottorete pubblica).
- In Auto-assign Public IP (Assegna automaticamente IP pubblico), scegli Enable (Abilita).
- Nella pagina Configure Security Group (Configura il gruppo di sicurezza), in Assign a security group (Assegna un gruppo di sicurezza), scegli Select an existing security group (Seleziona un gruppo di sicurezza esistente). A questo punto, seleziona il gruppo di sicurezza predefinito.
- Dopo aver selezionato Launch (Avvia), scegli una coppia di chiavi esistente o creane una.

Seleziona le impostazioni predefinite per tutte le altre opzioni.

7. Dopo l'esecuzione dell'istanza EC2, disabilita source/destination i controlli. Per informazioni, consulta la sezione [Disabling Source/Destination check](#) nella Amazon VPC User Guide.
8. Crea una tabella di routing nel VPC. Aggiungi alla tabella di routing una regola che consente di instradare il traffico destinato a Internet al server proxy. Associa la tabella di routing alla sottorete privata. Ciò è necessario affinché le richieste in uscita dalle istanze della sottorete privata, dove vengono CodeBuild eseguite, vengano sempre instradate attraverso il server proxy.

Installa e configura un server proxy

Esistono vari server proxy tra cui scegliere. Qui viene utilizzato un server proxy open source, Squid, per dimostrare come funziona AWS CodeBuild in un server proxy. Gli stessi concetti sono applicabili ad altri server proxy.

Per installare Squid, utilizza un repository yum eseguendo i seguenti comandi:

```
sudo yum update -y
sudo yum install -y squid
```

Dopo aver installato Squid, modifica il `squid.conf` utilizzando le istruzioni riportate più avanti in questo argomento.

Configura Squid per il traffico HTTPS

Per HTTPS, il traffico HTTP viene incapsulato in una connessione Transport Layer Security (TLS). Squid utilizza una funzionalità chiamata [SslPeekAndSplice](#) a recuperare l'indicazione del nome del server (SNI) dall'iniziazione TLS che contiene l'host Internet richiesto. In questo modo, Squid non

è obbligato a decrittografare il traffico HTTPS. Per abilitarlo SslPeekAndSplice, Squid richiede un certificato. Crea il certificato utilizzando OpenSSL:

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

Per HTTP, la configurazione di Squid non è richiesta. Da tutti i messaggi di richiesta HTTP/1.1 è possibile recuperare il campo dell'intestazione dell'host, che indica l'host Internet richiesto.

Esegui CodeBuild in un server proxy esplicito

Per l'esecuzione AWS CodeBuild in un server proxy esplicito, è necessario configurare il server proxy per consentire o negare il traffico da e verso siti esterni, quindi configurare le HTTP_PROXY variabili e di ambiente. HTTPS_PROXY

Argomenti

- [Configura Squid come server proxy esplicito](#)
- [Crea un progetto CodeBuild](#)
- [Esempio di file squid.conf del server proxy esplicito](#)

Configura Squid come server proxy esplicito

Per configurare il server proxy Squid in modo che sia esplicito, devi apportare al file `/etc/squid/squid.conf` le seguenti modifiche:

- Rimuovi le seguenti regole della lista di controllo accessi (ACL) predefinita.

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
```



```
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Sostituisci le regole ACL predefinite rimosse con quelle riportate di seguito. La prima riga consente le richieste dal VPC. Le due righe successive garantiscono al server proxy l'accesso alla destinazione URLs che potrebbe essere utilizzata da AWS CodeBuild Modifica l'espressione regolare nell'ultima riga per specificare i bucket S3 o un CodeCommit repository in una regione. AWS Esempio:

- Se la tua fonte è Amazon S3, usa il comando `acl download_src dstdom_regex .*s3\.us-west-1\.amazonaws\.com` per concedere l'accesso ai bucket S3 nella regione. `us-west-1`
- Se la tua fonte è AWS CodeCommit, usa `git-codecommit.<your-region>.amazonaws.com` per aggiungere una AWS regione a un elenco consentito.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from
Amazon S3 or CodeCommit
```

- Sostituisci `http_access allow localnet` con i seguenti:

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Se desideri che la compilazione carichi log e artefatti, esegui una delle operazioni seguenti:

1. Prima dell'istruzione `http_access deny all`, inserisci le istruzioni riportate di seguito, Consentono l'accesso CodeBuild a CloudWatch e l'accesso ad Amazon S3. L'accesso a CloudWatch è necessario per CodeBuild poter creare CloudWatch registri. L'accesso ad Amazon S3 è necessario per il caricamento di artefatti e la memorizzazione nella cache di Amazon S3.

```
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
```

```
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

- Dopo il salvataggio, esegui il comando `sequentesquid.conf`:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Aggiungi proxy al file di specifiche di compilazione. Per ulteriori informazioni, consulta [Sintassi buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
      - command
```

Note

Se ricevi un errore di `RequestError timeout`, consulta [RequestError errore di timeout durante l'esecuzione CodeBuild in un server proxy](#).

Per ulteriori informazioni, consulta [Esempio di file squid.conf del server proxy esplicito](#) più avanti in questo argomento.

Crea un progetto CodeBuild

Per eseguirlo AWS CodeBuild con il tuo server proxy esplicito, imposta le sue variabili `HTTP_PROXY` e di `HTTPS_PROXY` ambiente con l'indirizzo IP privato dell'istanza EC2 che hai creato per il tuo server proxy e la porta 3128 a livello di progetto. L'indirizzo IP privato è simile a questo `http://your-ec2-private-ip-address:3128`. Per ulteriori informazioni, consultare [Crea un progetto di compilazione in AWS CodeBuild](#) e [Modifica le impostazioni del progetto di compilazione in AWS CodeBuild](#).

Per visualizzare il log degli accessi del proxy Squid, utilizza il comando seguente:

```
sudo tail -f /var/log/squid/access.log
```

Esempio di file **squid.conf** del server proxy esplicito

Di seguito è riportato un esempio di file `squid.conf` configurato per un server proxy esplicito.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
# add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com    #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80  # http
acl Safe_ports port 21  # ftp
acl Safe_ports port 443  # https
acl Safe_ports port 70  # gopher
acl Safe_ports port 210  # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280  # http-mgmt
acl Safe_ports port 488  # gss-http
acl Safe_ports port 591  # filemaker
acl Safe_ports port 777  # multiling http
acl CONNECT method CONNECT
#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

```
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
# Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
# And finally deny all other access to this proxy
http_access deny all
# Squid normally listens to port 3128
http_port 3128
# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

Esegui CodeBuild in un server proxy trasparente

Per funzionare AWS CodeBuild in un server proxy trasparente, è necessario configurare il server proxy con l'accesso ai siti Web e ai domini con cui interagisce.

Argomenti

- [Configura di Squid come server proxy trasparente](#)

- [Crea un progetto CodeBuild](#)

Configura di Squid come server proxy trasparente

Per configurare un server proxy in modo che sia trasparente, deve poter accedere ai domini e ai siti web a cui desideri accedere. Per funzionare AWS CodeBuild con un server proxy trasparente, è necessario concedergli l'accesso. `amazonaws.com` È inoltre necessario concedere l'accesso agli CodeBuild usi di altri siti Web. Questi variano a seconda di come crei i tuoi CodeBuild progetti. I siti Web di esempio sono quelli per repository come Bitbucket GitHub, Yum e Maven. Per concedere a Squid l'accesso a domini e siti Web specifici, aggiorna il file `squid.conf` utilizzando un comando simile a quello riportato di seguito. Questo comando di esempio consente l'accesso a `amazonaws.com`, `github.com` e `bitbucket.com` e può essere modificato in modo da garantire l'accesso ad altri siti Web.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
EOF
```

Le richieste in entrata da istanze nella sottorete privata devono essere reindirizzate alle porte di Squid. Squid è in ascolto sulla porta 3129 per il traffico HTTP (anziché 80) e 3130 per il traffico HTTPS (anziché 443). Utilizza il comando iptables per instradare il traffico:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

Crea un progetto CodeBuild

Dopo aver configurato il server proxy, puoi utilizzarlo AWS CodeBuild in una sottorete privata senza ulteriori configurazioni. Ogni richiesta HTTP e HTTPS passa attraverso il server proxy pubblico. Per visualizzare il log degli accessi del proxy Squid, utilizza il comando seguente:

```
sudo tail -f /var/log/squid/access.log
```

Esegui un programma di gestione dei pacchetti e di altri strumenti in un server proxy

Utilizzare le seguenti procedure per eseguire un gestore di pacchetti e altri strumenti in un server proxy.

Per eseguire uno strumento, ad esempio un gestore di pacchetti, in un server proxy

1. Aggiungi lo strumento all'elenco di elementi consentiti nel server proxy inserendo le istruzioni nel file `squid.conf`.
2. Aggiungi una riga al file di specifiche di compilazione in modo che punti all'endpoint privato del server proxy.

Gli esempi riportati di seguito spiegano come eseguire questa operazione per `apt-get`, `curl` e `maven`. Se utilizzi uno strumento diverso, si applicano gli stessi principi. Aggiungilo a un elenco di autorizzazioni nel `squid.conf` file e aggiungi un comando al tuo file `buildspec` per identificare l'CodeBuild endpoint del tuo server proxy.

Per eseguire **apt-get** in un server proxy

1. Aggiungi le seguenti istruzioni al file `squid.conf` per inserire `apt-get` nell'elenco di elementi consentiti nel server proxy. Le prime tre righe consentono l'esecuzione nell'`apt-get` ambiente di compilazione.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Aggiungi l'istruzione riportata di seguito al file di specifiche di compilazione, in modo che i comandi `apt-get` cerchino la configurazione proxy in `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

Per eseguire **curl** in un server proxy

1. Aggiungi quanto segue al file `squid.conf` per inserire `curl` in un elenco di elementi consentiti nell'ambiente di compilazione.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a webiste. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Aggiungi l'istruzione riportata di seguito al file di specifiche di compilazione, in modo che `curl` acceda al sito Web aggiunto a `squid.conf` utilizzando il server proxy privato. In questo esempio, il sito Web è `google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

Per eseguire **maven** in un server proxy

1. Aggiungi quanto segue al file `squid.conf` per inserire maven in un elenco di elementi consentiti nell'ambiente di compilazione.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Aggiungi al file di specifiche di compilazione l'istruzione seguente.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

CloudFormation Modello VPC

CloudFormation consente di creare e fornire implementazioni di AWS infrastrutture in modo prevedibile e ripetuto, utilizzando file modello per creare ed eliminare una raccolta di risorse insieme come una singola unità (uno stack). Per ulteriori informazioni, consulta la [Guida per l'utente CloudFormation](#).

Di seguito è riportato un modello CloudFormation YAML per la configurazione di un VPC da utilizzare. AWS CodeBuild [Questo file è disponibile anche in samples.zip](#).

Description: This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

Parameters:

EnvironmentName:

Description: An environment name that is prefixed to resource names
Type: String

VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC
Type: String

Default: 10.192.0.0/16

PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

Type: String

Default: 10.192.10.0/24

PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

Type: String

Default: 10.192.11.0/24

PrivateSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 10.192.20.0/24

PrivateSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 10.192.21.0/24

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCIDR

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: !Ref EnvironmentName

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:

Tags:

- Key: Name

Value: !Ref EnvironmentName

```
InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC

PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet1CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet2CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PrivateSubnet1CIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
```

```
    CidrBlock: !Ref PrivateSubnet2CIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet (AZ2)

NatGateway1EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc

NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc

NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1

NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Routes

DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
```

```
    GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2

PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ1)

DefaultPrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ2)
```

```
DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "no-ingress-sg"
    GroupDescription: "Security group with no ingress rule"
    VpcId: !Ref VPC

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1

  PublicSubnet2:
    Description: A reference to the public subnet in the 2nd Availability Zone
    Value: !Ref PublicSubnet2

  PrivateSubnet1:
    Description: A reference to the private subnet in the 1st Availability Zone
    Value: !Ref PrivateSubnet1
```

PrivateSubnet2:

Description: A reference to the private subnet in the 2nd Availability Zone

Value: !Ref PrivateSubnet2

NoIngressSecurityGroup:

Description: Security group with no ingress rule

Value: !Ref NoIngressSecurityGroup

Registrazione e monitoraggio AWS CodeBuild

La registrazione e il monitoraggio sono elementi importanti per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS CodeBuild AWS soluzioni esistenti. È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce i seguenti strumenti per monitorare le CodeBuild risorse e le build e per rispondere a potenziali incidenti.

Argomenti

- [Registra le chiamate AWS CodeBuild API con AWS CloudTrail](#)
- [Il monitor CodeBuild si costruisce con CloudWatch](#)

Registra le chiamate AWS CodeBuild API con AWS CloudTrail

AWS CodeBuild è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in CodeBuild. CloudTrail acquisisce tutte le chiamate API relative CodeBuild agli eventi, incluse le chiamate dalla CodeBuild console e le chiamate in codice a. CodeBuild APIs Se crei un trail, puoi abilitare la consegna continua di CloudTrail eventi a un bucket S3, inclusi gli eventi per. CodeBuild Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata CodeBuild, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la [Guida AWS CloudTrail per l'utente](#).

Argomenti

- [AWS CodeBuild Informazioni in CloudTrail](#)
- [Informazioni sulle voci dei file di AWS CodeBuild registro](#)

AWS CodeBuild Informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in CodeBuild, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi

recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con cronologia degli CloudTrail eventi](#) nella Guida AWS CloudTrail per l'utente.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per CodeBuild, crea un percorso. Un trail consente di CloudTrail inviare file di registro a un bucket S3. Per impostazione predefinita, quando crei un trail nella console, il trail è valido per tutte le Regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di registro al bucket S3 specificato. Puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Tutte CodeBuild le azioni vengono registrate CloudTrail e documentate nell'[CodeBuild API Reference](#). Ad esempio, le chiamate alle azioni CreateProject (in the AWS CLI, create-project), StartBuild (in AWS CLI, start-project) e UpdateProject (in the AWS CLI, update-project) generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'[elemento CloudTrail userIdentity nella Guida](#) per l'AWS CloudTrail utente.

Informazioni sulle voci dei file di AWS CodeBuild registro

Un trail è una configurazione che consente la consegna di eventi come file di registro a un bucket S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni

sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Note

Per proteggere le informazioni sensibili, nei CodeBuild log sono nascosti i seguenti elementi:

- AWS chiave IDs di accesso. Per ulteriori informazioni, consulta [Managing Access Keys for IAM Users](#) nella Guida per l'AWS Identity and Access Management utente.
- Stringhe specificate utilizzando l'archivio parametri. Per ulteriori informazioni, consulta la [procedura dettagliata della console di Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.
- Stringhe specificate utilizzando. Gestione dei segreti AWS Per ulteriori informazioni, consulta [Gestione delle chiavi](#).

L'esempio seguente mostra una voce di CloudTrail registro che dimostra la creazione di un progetto di compilazione in CodeBuild

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "FederatedUser",
    "principalId": "account-ID:user-name",
    "arn": "arn:aws:sts::account-ID:federated-user/user-name",
    "accountId": "account-ID",
    "accessKeyId": "access-key-ID",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-09-06T17:59:10Z"
      },
      "sessionIssuer": {
        "type": "IAMUser",
        "principalId": "access-key-ID",
        "arn": "arn:aws:iam::account-ID:user/user-name",
        "accountId": "account-ID",
        "userName": "user-name"
      }
    }
  }
}
```

```
},
"eventTime": "2016-09-06T17:59:11Z",
"eventSource": "codebuild.amazonaws.com",
"eventName": "CreateProject",
"awsRegion": "region-ID",
"sourceIPAddress": "127.0.0.1",
"userAgent": "user-agent",
"requestParameters": {
  "awsActId": "account-ID"
},
"responseElements": {
  "project": {
    "environment": {
      "image": "image-ID",
      "computeType": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "name": "codebuild-demo-project",
    "description": "This is my demo project",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-  
project:project-ID",
    "encryptionKey": "arn:aws:kms:region-ID:key-ID",
    "timeoutInMinutes": 10,
    "artifacts": {
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
      "type": "S3",
      "packaging": "ZIP",
      "outputName": "MyOutputArtifact.zip"
    },
    "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
    "lastModified": "Sep 6, 2016 10:59:11 AM",
    "source": {
      "type": "GITHUB",
      "location": "https://github.com/my-repo.git"
    },
    "created": "Sep 6, 2016 10:59:11 AM"
  }
},
"requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
"eventID": "581f7dd1-8d2e-40b0-aeee-0dbf7EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "account-ID"
}
```

Il monitor CodeBuild si costruisce con CloudWatch

Puoi usare Amazon CloudWatch per controllare le tue build, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario. È possibile monitorare le compilazioni su due livelli:

A livello di progetto

Queste metriche si riferiscono a tutte le build del progetto specificato. Per visualizzare i parametri di un progetto, specificare il valore di `ProjectName` per la dimensione su CloudWatch.

AWS livello di account

Queste metriche si riferiscono a tutte le build di un account. Per visualizzare i parametri a livello dell'account AWS , non inserire una dimensione su CloudWatch. Le metriche sull'utilizzo delle risorse di compilazione non sono disponibili a livello di account. AWS

CloudWatch le metriche mostrano il comportamento delle tue build nel tempo. Ad esempio, è possibile monitorare:

- Quante build sono state tentate in un progetto di compilazione o in un account nel tempo. AWS
- Quante build hanno avuto successo in un progetto di compilazione o in un AWS account nel tempo.
- Quante build non sono riuscite in un progetto di compilazione o in un AWS account nel corso del tempo.
- Quanto tempo CodeBuild impiegato nel tempo per eseguire le build in un progetto di compilazione o in un AWS account.
- Sviluppa l'utilizzo delle risorse per una build o per un intero progetto di compilazione. Le metriche di utilizzo delle risorse di compilazione includono metriche come l'utilizzo della CPU, della memoria e dello storage.

Per ulteriori informazioni, consulta [Visualizza le CodeBuild metriche](#).

CodeBuild CloudWatch metriche

Le seguenti metriche possono essere monitorate per AWS account o progetto di build. Per ulteriori informazioni sull'utilizzo di CloudWatch with CodeBuild, consulta. [Il monitor CodeBuild si costruisce con CloudWatch](#)

BuildDuration

Misura la durata della fase BUILD della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

Compilazioni

Misura il numero di compilazioni attivate.

Unità: numero

CloudWatch Statistiche valide: somma

DownloadSourceDuration

Misura la durata della fase DOWNLOAD_SOURCE della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

Durata

Misura la durata nel tempo di tutte le compilazioni.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

FailedBuilds

Misura il numero di build non riuscite a causa di un errore del client o di un timeout.

Unità: numero

CloudWatch Statistiche valide: somma

FinalizingDuration

Misura la durata della fase FINALIZING della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

InstallDuration

Misura la durata della fase INSTALL della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

PostBuildDuration

Misura la durata della fase POST_BUILD della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

PreBuildDuration

Misura la durata della fase PRE_BUILD della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

ProvisioningDuration

Misura la durata della fase PROVISIONING della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

QueuedDuration

Misura la durata della fase QUEUED della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

SubmittedDuration

Misura la durata della fase SUBMITTED della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

SucceededBuilds

Misura il numero di compilazioni riuscite.

Unità: numero

CloudWatch Statistiche valide: somma

UploadArtifactsDuration

Misura la durata della fase UPLOAD_ARTIFACTS della compilazione.

Unità: secondi

CloudWatch Statistiche valide: media (consigliata), massima, minima

CodeBuild CloudWatch metriche di utilizzo delle risorse

Note

CodeBuild le metriche di utilizzo delle risorse sono disponibili solo nelle seguenti aree:

- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Seoul)
- Regione Asia Pacifico (Mumbai)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Canada (Centrale)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)

- Regione Europa (Londra)
- Regione Europa (Parigi)
- Regione Sud America (San Paolo)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Regione Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)

È possibile tenere traccia delle seguenti metriche di utilizzo delle risorse. Per ulteriori informazioni sull'utilizzo CloudWatch con CodeBuild, vedere. [Il monitor CodeBuild si costruisce con CloudWatch](#)

CPUUtilized

Il numero di unità CPU di elaborazione allocata utilizzate dal contenitore di compilazione.

Unità: unità CPU

CloudWatch Statistiche valide: media (consigliata), massima, minima

CPUUtilizedPercentuale

La percentuale di elaborazione allocata utilizzata dal contenitore di compilazione.

Unità: percentuale

CloudWatch Statistiche valide: media (consigliata), massima, minima

MemoryUtilized

Il numero di megabyte di memoria utilizzati dal contenitore di build.

Unità: megabyte

CloudWatch Statistiche valide: media (consigliata), massima, minima

MemoryUtilizedPercent

La percentuale di memoria allocata utilizzata dal contenitore di build.

Unità: percentuale

CloudWatch Statistiche valide: media (consigliata), massima, minima

StorageReadBytes

La velocità di lettura dello storage utilizzata dal contenitore di compilazione.

Unità: byte/secondo

CloudWatch Statistiche valide: media (consigliata), massima, minima

StorageWriteBytes

La velocità di scrittura dello storage utilizzata dal contenitore di compilazione.

Unità: byte/secondo

CloudWatch Statistiche valide: media (consigliata), massima, minima

CodeBuild CloudWatch dimensioni

CodeBuild fornisce le seguenti dimensioni CloudWatch metriche. Se nessuna di queste viene specificata, le metriche si riferiscono all'account corrente AWS .

BuildId, BuildNumber, ProjectName

Vengono fornite le metriche per l'identificatore di build, il numero di build e il nome del progetto.

ProjectName

Le metriche vengono fornite per il nome di un progetto.

CodeBuild CloudWatch allarmi

Puoi usare la CloudWatch console per creare allarmi basati su CodeBuild metriche in modo da poter reagire se qualcosa va storto con le tue build. Le due metriche più utili per gli allarmi sono descritte nei punti seguenti. Per ulteriori informazioni sull'utilizzo di CloudWatch with CodeBuild, consulta. [|| monitor CodeBuild si costruisce con CloudWatch](#)

- **FailedBuild.** È possibile creare un allarme che viene attivato quando viene rilevato un certo numero di build non riuscite entro un numero predeterminato di secondi. In CloudWatch, si specifica il numero di secondi e il numero di build fallite che attivano un allarme.

- **Duration.** È possibile creare un allarme che viene attivato quando una compilazione richiede più tempo del previsto. Puoi specificare quanti secondi devono trascorrere prima che si attivi un allarme tra l'avvio della compilazione e il suo completamento.

Per informazioni su come creare allarmi per le CodeBuild metriche, consulta [Monitora CodeBuild le build con allarmi CloudWatch](#). Per ulteriori informazioni sugli allarmi, consulta la sezione [Creazione di CloudWatch allarmi Amazon](#) nella Amazon CloudWatch User Guide.

Visualizza le CodeBuild metriche

AWS CodeBuild monitora le funzioni per tuo conto e riporta i parametri tramite Amazon. CloudWatch. Questi parametri includono il numero di compilazioni totali, compilazioni non riuscite, compilazioni riuscite e la durata delle compilazioni.

Puoi utilizzare la CodeBuild console o la CloudWatch console per monitorare le metriche di CodeBuild. Le seguenti procedure mostrano come visualizzare le metriche.

Argomenti

- [Visualizza le metriche di compilazione \(console\) CodeBuild](#)
- [Visualizza i parametri di compilazione \(CloudWatch console Amazon\)](#)

Visualizza le metriche di compilazione (console) CodeBuild

Note

Non puoi personalizzare le metriche o i grafici utilizzati per visualizzarli nella console. CodeBuild. Se desideri personalizzare lo schermo, usa la CloudWatch console Amazon per visualizzare i parametri di build.

Metriche a livello di account

Per visualizzare le metriche a livello di account AWS

1. [Accedi Console di gestione AWS e apri la AWS CodeBuild console su codebuild/home. https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Nel riquadro di navigazione, selezionare Account metrics (Parametri account).

Metriche a livello di progetto

Per visualizzare le metriche a livello di progetto

1. [Accedi Console di gestione AWS e apri la AWS CodeBuild console su codebuild/home. https://console.aws.amazon.com/codesuite/](https://console.aws.amazon.com/codesuite/)
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Nell'elenco di progetti di compilazione, nella colonna Name (Nome), selezionare il progetto in cui visualizzare i parametri.
4. Seleziona la scheda Parametri.

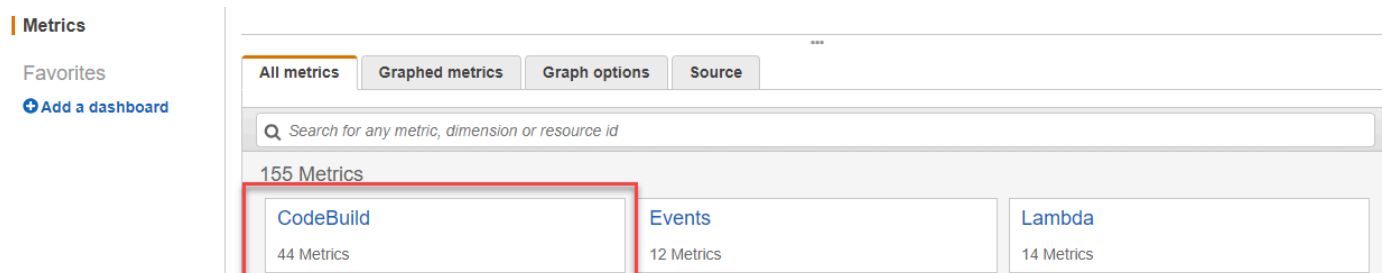
Visualizza i parametri di compilazione (CloudWatch console Amazon)

Puoi personalizzare le metriche e i grafici utilizzati per visualizzarle con la console. CloudWatch

Metriche a livello di account

Per visualizzare le metriche a livello di account

1. Accedi a Console di gestione AWS e apri la console all' CloudWatch indirizzo. <https://console.aws.amazon.com/cloudwatch/>
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nella scheda Tutti i parametri, scegli CodeBuild.

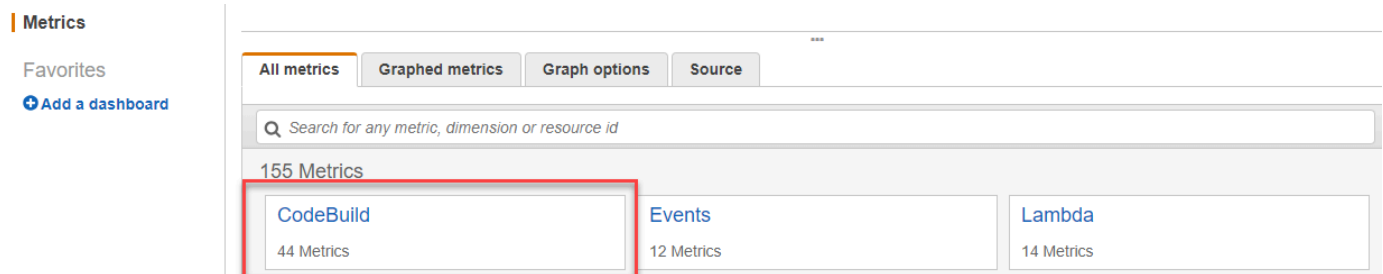


4. Seleziona Account Metrics (Parametri account).
5. Selezionare uno o più progetti e parametri. Per ogni progetto, puoi scegliere le metriche SucceededBuilds, FailedBuilds, Builds e Duration. Tutte le combinazioni di progetti e parametri selezionate sono visualizzate nel grafico sulla pagina.

Metriche a livello di progetto

Per visualizzare le metriche a livello di progetto

1. Accedi a Console di gestione AWS e apri la console all' CloudWatch indirizzo. <https://console.aws.amazon.com/cloudwatch/>
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nella scheda Tutti i parametri, scegli CodeBuild.



4. Selezionare By project (Per progetto).
5. Selezionare una o più combinazioni di progetti e parametri. Per ogni progetto, puoi scegliere le metriche SucceededBuilds, FailedBuilds, Builds e Duration. Tutte le combinazioni di progetti e parametri selezionate sono visualizzate nel grafico sulla pagina.
6. (Facoltativo), è possibile personalizzare i parametri e i grafici. Ad esempio, dall'elenco a discesa nella colonna Statistica, puoi scegliere una statistica diversa da visualizzare. Oppure, dal menu a discesa nella colonna Period (Periodo), è possibile selezionare di utilizzare un periodo di tempo diverso per monitorare i parametri.

Per ulteriori informazioni, consulta [Graph metrics](#) e [View available metrics](#) nella Amazon CloudWatch User Guide.

Visualizza le CodeBuild metriche sull'utilizzo delle risorse

AWS CodeBuild monitora l'utilizzo delle risorse dello sviluppo per tuo conto e riporta i parametri tramite Amazon. CloudWatch Questi includono parametri come l'utilizzo della CPU, della memoria e dello storage.

Note

CodeBuild le metriche di utilizzo delle risorse vengono registrate solo per le build che vengono eseguite per più di un minuto.

È possibile utilizzare la CodeBuild console o la console per monitorare i CloudWatch parametri di utilizzo delle risorse per. CodeBuild

Note

CodeBuild le metriche di utilizzo delle risorse sono disponibili solo nelle seguenti regioni:

- Regione Asia Pacifico (Tokyo)
- Regione Asia Pacifico (Seoul)
- Regione Asia Pacifico (Mumbai)
- Regione Asia Pacifico (Singapore)
- Regione Asia Pacifico (Sydney)
- Regione Canada (Centrale)
- Regione Europa (Francoforte)
- Regione Europa (Irlanda)
- Regione Europa (Londra)
- Regione Europa (Parigi)
- Regione Sud America (San Paolo)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Regione Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)

Le seguenti procedure mostrano come accedere alle metriche di utilizzo delle risorse.

Argomenti

- [Accedi ai parametri di utilizzo delle risorse \(console\) CodeBuild](#)
- [Accedi ai parametri di utilizzo delle risorse \(console Amazon CloudWatch \)](#)

Accedi ai parametri di utilizzo delle risorse (console) CodeBuild

Note

Non puoi personalizzare le metriche o i grafici utilizzati per visualizzarli nella console. CodeBuild Se desideri personalizzare lo schermo, usa la CloudWatch console Amazon per visualizzare i parametri di build.

Metriche di utilizzo delle risorse a livello di progetto

Per accedere alle metriche di utilizzo delle risorse a livello di progetto

1. [Accedi Console di gestione AWS e apri la console all'indirizzo codebuild/home. AWS CodeBuild](https://console.aws.amazon.com/codesuite/)
<https://console.aws.amazon.com/codesuite/>
2. Nel riquadro di navigazione, scegliere Build projects (Progetti di compilazione).
3. Nell'elenco dei progetti di compilazione, nella colonna Nome, scegli il progetto per cui desideri visualizzare le metriche di utilizzo.
4. Seleziona la scheda Parametri. Le metriche sull'utilizzo delle risorse vengono visualizzate nella sezione Metriche sull'utilizzo delle risorse.
5. Per visualizzare le metriche di utilizzo delle risorse a livello di progetto nella CloudWatch console, scegli Visualizza nella sezione Metriche sull'utilizzo delle risorse. CloudWatch

Metriche di utilizzo delle risorse a livello di build

Per accedere alle metriche di utilizzo delle risorse a livello di build

1. [Accedi Console di gestione AWS e apri la console all'indirizzo codebuild/home. AWS CodeBuild](https://console.aws.amazon.com/codesuite/)
<https://console.aws.amazon.com/codesuite/>
2. Nel riquadro di navigazione, selezionare Build history (Cronologia build).
3. Nell'elenco delle build, nella colonna Build run, scegli la build per cui desideri visualizzare le metriche di utilizzo.
4. Scegli la scheda Utilizzo delle risorse.
5. Per visualizzare le metriche di utilizzo delle risorse a livello di build nella CloudWatch console, scegli Visualizza CloudWatch nella sezione Metriche sull'utilizzo delle risorse.

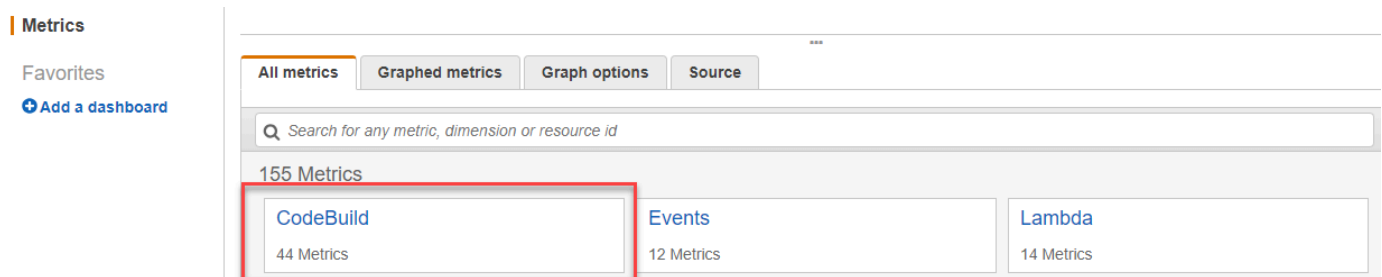
Accedi ai parametri di utilizzo delle risorse (console Amazon CloudWatch)

La CloudWatch console Amazon può essere utilizzata per accedere ai parametri di utilizzo CodeBuild delle risorse.

Metriche di utilizzo delle risorse a livello di progetto

Per accedere alle metriche di utilizzo delle risorse a livello di progetto

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CloudWatch <https://console.aws.amazon.com/cloudwatch/>
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nella scheda Tutti i parametri, scegli CodeBuild.



4. Selezionare By project (Per progetto).
5. Scegli una o più combinazioni di progetti e metriche da aggiungere al grafico. Tutte le combinazioni di progetti e parametri selezionate sono visualizzate nel grafico sulla pagina.
6. (Facoltativo) Puoi personalizzare le metriche e i grafici dalla scheda Metriche grafiche. Ad esempio, dall'elenco a discesa nella colonna Statistiche, puoi scegliere una statistica diversa da visualizzare. Oppure, dal menu a discesa nella colonna Period (Periodo), è possibile selezionare di utilizzare un periodo di tempo diverso per monitorare i parametri.

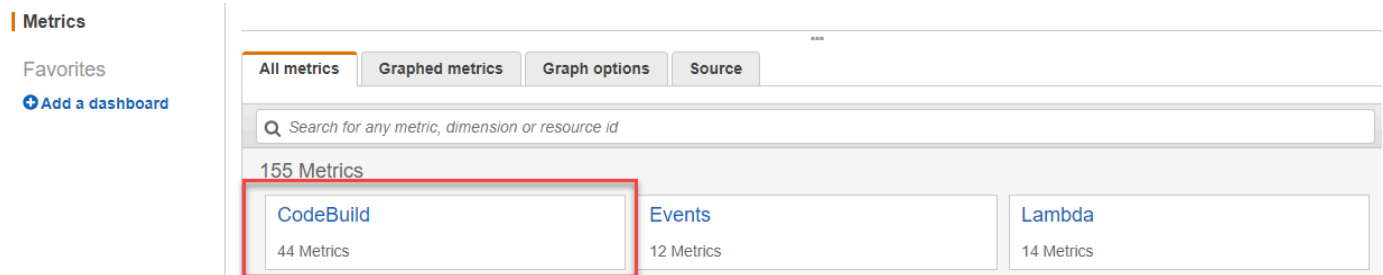
Per ulteriori informazioni, consulta la sezione [Rappresentazione grafica delle metriche](#) e [Visualizzazione delle metriche disponibili](#) nella Amazon CloudWatch User Guide.

Metriche di utilizzo delle risorse a livello di build

Per accedere alle metriche di utilizzo delle risorse a livello di build

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. CloudWatch <https://console.aws.amazon.com/cloudwatch/>
2. Nel riquadro di navigazione, seleziona Parametri.

3. Nella scheda Tutti i parametri, scegli CodeBuild.



4. Scegli BuildId,, BuildNumber. ProjectName

5. Scegli una o più combinazioni di build e metriche da aggiungere al grafico. Tutte le combinazioni di build e metriche selezionate vengono visualizzate nel grafico della pagina.
6. (Facoltativo) Puoi personalizzare le metriche e i grafici dalla scheda Metriche grafiche. Ad esempio, dall'elenco a discesa nella colonna Statistiche, puoi scegliere una statistica diversa da visualizzare. Oppure, dal menu a discesa nella colonna Period (Periodo), è possibile selezionare di utilizzare un periodo di tempo diverso per monitorare i parametri.

Per ulteriori informazioni, consulta la sezione [Rappresentazione grafica delle metriche](#) e [Visualizzazione delle metriche disponibili](#) nella Amazon CloudWatch User Guide.

Monitora CodeBuild le build con allarmi CloudWatch

Puoi creare un CloudWatch allarme per le tue build. Un allarme è collegato a un singolo parametro per tutto il tempo specificato ed esegue una o più azioni in base alla relazione tra il valore del parametro e una soglia impostata, su più intervalli predefiniti. Utilizzando la funzionalità di CloudWatch allarme nativa, puoi specificare qualsiasi azione supportata dal CloudWatch superamento di una soglia. Ad esempio, puoi specificare che venga inviata una notifica Amazon SNS quando più di tre build del tuo account falliscono entro quindici minuti.

Per creare un CloudWatch allarme per una metrica CodeBuild

1. Accedi a Console di gestione AWS e apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nel pannello di navigazione, seleziona Alarms (Allarmi).
3. Scegli Create Alarm (Crea allarme).
4. In CloudWatch Metriche per categoria, scegli CodeBuildMetriche. Se è già noto che i parametri desiderati sono solo a livello di progetto, selezionare By Project (Per progetto). Se è già noto

che i parametri desiderati sono solo a livello di account, selezionare Account Metrics (Parametri account).

5. In Create Alarm (Crea allarme), selezionare Select Metric (Seleziona parametro), se non è già selezionato.
6. Selezionare un parametro per il quale si intende creare un allarme. Le opzioni sono By Project (Per progetto) o Account Metrics (Parametri account).
7. Selezionare Next (Avanti) o Define Alarm (Definisci allarme) e creare quindi l'allarme. Per ulteriori informazioni, consulta [Creazione di CloudWatch allarmi Amazon](#) nella Amazon CloudWatch User Guide. Per ulteriori informazioni sulla configurazione delle notifiche di Amazon SNS quando viene attivato un allarme, consulta [Configurare le notifiche di Amazon SNS nella Amazon SNS Developer Guide](#).
8. Scegli Crea allarme.

Sicurezza in AWS CodeBuild

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei [programmi di conformitàAWS](#). Per ulteriori informazioni sui programmi di conformità applicabili AWS CodeBuild, consulta [AWS Services in Scope by Compliance Program](#).
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo CodeBuild. Negli argomenti seguenti viene illustrato come eseguire la configurazione CodeBuild per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere CodeBuild le tue risorse.

Argomenti

- [Protezione dei dati in AWS CodeBuild](#)
- [Gestione delle identità e degli accessi in AWS CodeBuild](#)
- [Convalida della conformità per AWS CodeBuild](#)
- [Resilienza in AWS CodeBuild](#)
- [Sicurezza dell'infrastruttura in AWS CodeBuild](#)
- [Accedi al tuo provider di origine in CodeBuild](#)
- [Prevenzione del confused deputy tra servizi](#)

Protezione dei dati in AWS CodeBuild

Il [modello di responsabilità AWS condivisa](#) di si applica alla protezione dei dati in AWS CodeBuild. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [AWS Modello di responsabilità condivisa e GDPR](#) nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori CodeBuild o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo

vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Per proteggere le informazioni sensibili, nei CodeBuild log sono nascoste le seguenti informazioni:

- Stringhe specificate utilizzando il Parameter Store nelle variabili di ambiente CodeBuild del progetto o nella sezione `env/parameter-store` buildspec. Per ulteriori informazioni, consulta la [procedura dettagliata della console Systems Manager Parameter Store e Systems Manager Parameter Store nella Guida](#) per l'utente di Amazon EC2 Systems Manager.
- Stringhe specificate utilizzando Gestione dei segreti AWS le variabili di ambiente CodeBuild del progetto o la sezione `buildspec. env/secrets-manager`. Per ulteriori informazioni, consulta [Gestione delle chiavi](#).

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog relativo al [modello di responsabilità condivisa AWS e GDPR](#) in AWS Security Blog.

Argomenti

- [Crittografia dei dati](#)
- [Gestione delle chiavi](#)
- [Privacy del traffico](#)

Crittografia dei dati

La crittografia è una parte importante della sicurezza. CodeBuild Alcuni tipi di crittografia, ad esempio quella per i dati in transito, vengono forniti per impostazione predefinita e l'utente non deve fare nulla. Altri tipi di crittografia, come quella per i dati inattivi, sono configurabili quando si crea un progetto o una compilazione.

- Crittografia dei dati inattivi: gli elementi della build, come la cache, i log, i file di dati non elaborati esportati nei report dei test e i risultati della build, vengono crittografati per impostazione predefinita utilizzando chiavi gestite da AWS. Se non desideri utilizzare queste chiavi KMS, devi creare e configurare una chiave gestita dal cliente. Per ulteriori informazioni, consulta [Creazione delle chiavi KMS](#) e [Concetti di AWS Key Management Service](#) nella Guida per l'utente di AWS Key Management Service.
- È possibile memorizzare l'identificatore della chiave AWS KMS CodeBuild utilizzata per crittografare l'elemento di output della build nella variabile di ambiente.

CODEBUILD_KMS_KEY_ID Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#)

- È possibile specificare una chiave gestita dal cliente quando si crea un progetto di compilazione. Per ulteriori informazioni, consulta [Set the Encryption Key Using the Console](#) e [Imposta la chiave di crittografia utilizzando la CLI](#).

I volumi Amazon Elastic Block Store della tua flotta di build sono crittografati per impostazione predefinita utilizzando Chiavi gestite da AWS.

- Crittografia dei dati in transito: tutte le comunicazioni tra i clienti CodeBuild CodeBuild e tra le relative dipendenze a valle sono protette mediante connessioni TLS firmate utilizzando il processo di firma Signature versione 4. Tutti gli CodeBuild endpoint utilizzano certificati SHA-256 gestiti da AWS Autorità di certificazione privata Per ulteriori informazioni consulta la pagina relativa al [processo di firma Signature Version 4](#) e la pagina [Che cos'è ACM PCA](#).
- Crittografia degli artefatti di compilazione: il ruolo di CodeBuild servizio associato al progetto di compilazione richiede l'accesso a una chiave KMS per crittografare gli artefatti di output della build. Per impostazione predefinita, CodeBuild utilizza un Chiave gestita da AWS per Amazon S3 nel tuo AWS account. Se non desideri utilizzarlo Chiave gestita da AWS, devi creare e configurare una chiave gestita dal cliente. Per ulteriori informazioni, consulta [Crittografa gli output di compilazione Creazione delle chiavi](#) nella Guida per gli AWS KMS sviluppatori.

Gestione delle chiavi

Puoi proteggere i tuoi contenuti da un uso non autorizzato tramite la crittografia. Archivia le tue chiavi di crittografia Gestione dei segreti AWS, quindi concedi al ruolo di CodeBuild servizio associato al progetto di compilazione l'autorizzazione a ottenere le chiavi di crittografia dal tuo account Secrets Manager. Per ulteriori informazioni, consulta [Crittografa gli output della build utilizzando una chiave gestita dal cliente](#), [Crea un progetto di compilazione in AWS CodeBuild](#), [Esegui AWS CodeBuild le build manualmente](#) e [Tutorial: archiviazione e recupero di un segreto](#).

Usa la variabile di CODEBUILD_KMS_KEY_ID ambiente in un comando build per ottenere l'identificatore della AWS KMS chiave. Per ulteriori informazioni, consulta [Variabili di ambiente degli ambienti di compilazione](#).

Puoi usare Secrets Manager per proteggere le credenziali di un registro privato che memorizza un'immagine Docker utilizzata per il tuo ambiente di runtime. Per ulteriori informazioni, consulta [Registro privato con Gestione dei segreti AWS esempio per CodeBuild](#).

Privacy del traffico

Puoi migliorare la sicurezza delle tue build configurando l'uso di un CodeBuild endpoint VPC di interfaccia. Per far ciò, non hai bisogno di un gateway Internet, di un dispositivo NAT o di un gateway privato virtuale. Inoltre, non è necessaria la configurazione PrivateLink, sebbene sia consigliata. Per ulteriori informazioni, consulta [Utilizzo degli endpoint VPC](#). Per ulteriori informazioni sugli PrivateLink endpoint VPC, consulta [AWS PrivateLink](#) e [Accesso ai AWS](#) servizi tramite PrivateLink.

Gestione delle identità e degli accessi in AWS CodeBuild

L'accesso a AWS CodeBuild richiede credenziali. Tali credenziali devono disporre delle autorizzazioni per accedere alle AWS risorse, come l'archiviazione e il recupero degli artefatti di build nei bucket S3 e la visualizzazione di Amazon Logs per le build. CloudWatch Le seguenti sezioni descrivono come utilizzare [AWS Identity and Access Management](#) (IAM) e come contribuire a proteggere l'accesso alle risorse: CodeBuild

Panoramica della gestione delle autorizzazioni di accesso alle risorse AWS CodeBuild

Ogni AWS risorsa è di proprietà di un AWS account e le autorizzazioni per creare o accedere a una risorsa sono regolate dalle politiche di autorizzazione. Un amministratore dell'account è in grado di collegare le policy relative alle autorizzazioni alle identità IAM (ovvero utenti, gruppi e ruoli).

Note

Un amministratore account (o un utente amministratore) è un utente con privilegi di amministratore. Per ulteriori informazioni, consulta [Best practice IAM](#) nella Guida per l'utente di IAM.

Quando concedi le autorizzazioni, devi specificare gli utenti che le riceveranno, le risorse a cui potranno accedere e le operazioni consentite su tali risorse.

Argomenti

- [AWS CodeBuild risorse e operazioni](#)
- [Informazioni sulla proprietà delle risorse](#)

- [Gestione dell'accesso alle risorse](#)
- [Specifica degli elementi delle policy: operazioni, effetti e principali](#)

AWS CodeBuild risorse e operazioni

Nel AWS CodeBuild, la risorsa principale è un progetto di compilazione. In una policy, devi utilizzare un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy. Le build sono anche risorse e sono ARNs associate ad esse. Per ulteriori informazioni, consulta [Amazon Resource Names \(ARN\) e AWS Service Namespaces](#) nel. Riferimenti generali di Amazon Web Services

Tipo di risorsa	Formato ARN
Progetto di compilazione	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :project/ <i>project-name</i>
Creazione	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :build/ <i>build-ID</i>
Gruppo di report	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report-group/ <i>report-group-name</i>
Report	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report/ <i>report-ID</i>
Parco istanze	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :fleet/ <i>fleet-ID</i>
Tutte le risorse CodeBuild	arn:aws:codebuild:*
Tutte CodeBuild le risorse di proprietà dell'account specifico nella AWS regione specificata	arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :*

 Important

Quando si utilizza la funzionalità di capacità riservata, i dati memorizzati nella cache delle istanze del parco istanze, inclusi i file di origine, i layer Docker e le directory memorizzate nella cache specificate nella buildspec, possono essere accessibili ad altri progetti all'interno dello stesso account. Questa funzionalità è preimpostata e consente ai progetti all'interno dello stesso account di condividere le istanze del parco istanze.

 Note

La maggior parte dei AWS servizi considera i due punti (:) o una barra (/) come lo stesso carattere in. ARNs Tuttavia, CodeBuild utilizza una corrispondenza esatta nei modelli e nelle regole delle risorse. Assicurati di utilizzare i caratteri ARN corretti durante la creazione di modelli di eventi, facendo in modo che corrispondano alla sintassi ARN nella risorsa.

Ad esempio, puoi indicare uno specifico progetto di build (*myBuildProject*) nella tua dichiarazione utilizzando il relativo ARN come segue:

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Per specificare tutte le risorse, o se un'azione API non supporta ARNs, utilizzate il carattere jolly (*) nell'Resource elemento come segue:

```
"Resource": "*"
```

Alcune azioni CodeBuild API accettano più risorse (ad esempio, BatchGetProjects). Per specificare più risorse in una singola istruzione, separare ARNs con virgole, come segue:

```
"Resource": [  
  "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",  
  "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"  
]
```

CodeBuild fornisce una serie di operazioni per utilizzare le CodeBuild risorse. Per un elenco, consulta [AWS CodeBuild riferimento alle autorizzazioni](#).

Informazioni sulla proprietà delle risorse

L' AWS account possiede le risorse create nell'account, indipendentemente da chi ha creato le risorse. In particolare, il proprietario della risorsa è l' AWS account dell'[entità principale](#) (ovvero l'account root, un utente o un ruolo IAM) che autentica la richiesta di creazione delle risorse. Negli esempi seguenti viene illustrato il funzionamento:

- Se utilizzi le credenziali dell'account root del tuo AWS account per creare una regola, quest'ultimo AWS è il proprietario della risorsa. CodeBuild
- Se crei un utente nel tuo AWS account e concedi le autorizzazioni per creare CodeBuild risorse a quell'utente, l'utente può creare CodeBuild risorse. Tuttavia, il tuo AWS account, a cui appartiene l'utente, possiede le CodeBuild risorse.
- Se crei un ruolo IAM nel tuo AWS account con le autorizzazioni per creare CodeBuild risorse, chiunque possa assumere il ruolo può creare CodeBuild risorse. Il tuo AWS account, a cui appartiene il ruolo, possiede le CodeBuild risorse.

Gestione dell'accesso alle risorse

La policy delle autorizzazioni descrive chi ha accesso a quali risorse.

Note

In questa sezione si esamina l'utilizzo di IAM in AWS CodeBuild. Non vengono fornite informazioni dettagliate sul servizio IAM. Per la documentazione di IAM completa, consulta [Che cos'è IAM?](#) nella Guida per l'utente di IAM. Per informazioni sulla sintassi delle policy IAM e le rispettive descrizioni, consulta [Riferimento alle policy IAM di AWS](#) nella Guida per l'utente di IAM.

Le policy collegate a un'identità IAM vengono definite policy basate su identità (policy IAM). Le politiche associate a una risorsa vengono chiamate politiche basate sulle risorse. CodeBuild supporta politiche basate sull'identità e politiche basate sulle risorse, in alcuni casi di sola lettura, ai fini della condivisione delle risorse tra account. APIs

Accesso sicuro ai bucket S3

Ti consigliamo vivamente di includere le seguenti autorizzazioni nel tuo ruolo IAM per verificare che il bucket S3 associato al tuo progetto sia di tua proprietà o di qualcuno di cui ti CodeBuild fidi. Queste autorizzazioni non sono incluse nelle politiche e nei ruoli AWS gestiti. Devi aggiungerle da solo.

- `s3:GetBucketAcl`
- `s3:GetBucketLocation`

Se il proprietario di un bucket S3 utilizzato dal tuo progetto cambia, devi verificare di essere ancora proprietario del bucket e, in caso contrario, aggiornare le autorizzazioni nel tuo ruolo IAM. Per ulteriori informazioni, consultare [Consenti agli utenti di interagire con CodeBuild](#) e [Consenti CodeBuild di interagire con altri servizi AWS](#).

Specifica degli elementi delle policy: operazioni, effetti e principali

Per ogni AWS CodeBuild risorsa, il servizio definisce un set di operazioni API. Per concedere le autorizzazioni per queste operazioni API, CodeBuild definisce una serie di azioni che è possibile specificare in una politica. Alcune operazioni API possono richiedere le autorizzazioni per più di un'azione al fine di eseguire l'operazione API. Per ulteriori informazioni, consultare [AWS CodeBuild risorse e operazioni](#) e [AWS CodeBuild riferimento alle autorizzazioni](#).

Di seguito sono elencati gli elementi di base di una policy:

- **Risorsa** - Usa un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy.
- **Azione**: si utilizzano parole chiave di azione per identificare le operazioni sulle risorse che si desidera consentire o negare. Ad esempio, l'autorizzazione `codebuild:CreateProject` fornisce all'utente le autorizzazioni per eseguire l'operazione `CreateProject`.
- **Effetto**: si specifica l'effetto, che può essere consentito o negato, quando l'utente richiede l'azione. **USE** non concedi esplicitamente (consenti) l'accesso a una risorsa, l'accesso viene implicitamente rifiutato. È anche possibile negare esplicitamente l'accesso a una risorsa. È possibile eseguire questa operazione per assicurarsi che un utente non possa accedere alla risorsa, anche se l'accesso viene concesso da un'altra policy.
- **Principio**: nelle politiche basate sull'identità (politiche IAM), l'utente a cui è associata la policy è il principale implicito. Per le policy basate sulle risorse, devi specificare utente, account, servizio o altre entità che desideri ricevano le autorizzazioni.

Per ulteriori informazioni sulla sintassi e le descrizioni delle policy IAM, consulta [AWS Riferimento alle policy IAM](#) nella Guida per l'utente di IAM.

Per una tabella che mostra tutte le azioni CodeBuild API e le risorse a cui si applicano, consulta la [AWS CodeBuild riferimento alle autorizzazioni](#)

Utilizzo di politiche basate sull'identità per AWS CodeBuild

Questo argomento fornisce esempi di policy basate su identità che illustrano come un amministratore account può associare policy di autorizzazione a identità IAM (ovvero utenti, gruppi e ruoli) e quindi concedere autorizzazioni per eseguire operazioni su risorse di AWS CodeBuild .

Important

Ti consigliamo di esaminare innanzitutto gli argomenti introduttivi che spiegano i concetti e le opzioni di base disponibili per gestire l'accesso alle tue risorse. CodeBuild Per ulteriori informazioni, consulta [Panoramica della gestione delle autorizzazioni di accesso alle risorse AWS CodeBuild](#).

Argomenti

- [Autorizzazioni necessarie per utilizzare la console AWS CodeBuild](#)
- [Autorizzazioni necessarie per connettersi AWS CodeBuild ad Amazon Elastic Container Registry](#)
- [Autorizzazioni necessarie per consentire alla AWS CodeBuild console di connettersi ai provider di origine](#)
- [AWS politiche gestite \(predefinite\) per AWS CodeBuild](#)
- [CodeBuild politiche e notifiche gestite](#)
- [CodeBuild aggiornamenti alle politiche AWS gestite](#)
- [Esempi di policy gestite dal cliente](#)

Segue un esempio di una policy di autorizzazioni che permette all'utente di ottenere informazioni sui progetti di compilazione solo nella regione us-east-2 per l'account 123456789012 per qualsiasi progetto di compilazione che inizia con il nome my:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Autorizzazioni necessarie per utilizzare la console AWS CodeBuild

Un utente che utilizza la AWS CodeBuild console deve disporre di un set minimo di autorizzazioni che gli consenta di descrivere altre AWS risorse per l' AWS account. Deve possedere le autorizzazioni dai seguenti servizi:

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (se stai archiviando il codice sorgente in un AWS CodeCommit repository)
- Amazon Elastic Container Registry (Amazon ECR) (se utilizzi un ambiente di compilazione basato su un'immagine Docker in un repository Amazon ECR)

Note

A partire dal 26 luglio 2022, la policy IAM predefinita è stata aggiornata. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per connettersi AWS CodeBuild ad Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (se utilizzi un ambiente di compilazione basato su un'immagine Docker in un repository Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Se crei una policy IAM più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto.

Autorizzazioni necessarie per connettersi AWS CodeBuild ad Amazon Elastic Container Registry

Al 26 luglio 2022, AWS CodeBuild ha aggiornato la sua politica IAM predefinita per l'autorizzazione Amazon ECR. Le seguenti autorizzazioni sono state rimosse dalla politica predefinita:

```
"ecr:PutImage",  
"ecr:InitiateLayerUpload",  
"ecr:UploadLayerPart",  
"ecr:CompleteLayerUpload"
```

Per CodeBuild i progetti creati prima del 26 luglio 2022, ti consigliamo di aggiornare la tua politica con la seguente politica di Amazon ECR:

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

Per ulteriori informazioni sull'aggiornamento della polizza, consulta [Consenti agli utenti di interagire con CodeBuild](#).

Autorizzazioni necessarie per consentire alla AWS CodeBuild console di connettersi ai provider di origine

La AWS CodeBuild console utilizza le seguenti azioni API per connettersi ai provider di origine (ad esempio, gli GitHub archivi).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Puoi associare i fornitori di sorgenti (come i GitHub repository) ai tuoi progetti di build utilizzando la AWS CodeBuild console. Per fare ciò, devi prima aggiungere le azioni API precedenti alle policy di accesso IAM associate all'utente che utilizzi per accedere alla AWS CodeBuild console.

Le operazioni API `ListConnectedOAuthAccounts`, `ListRepositories` e `PersistOAuthToken` non devono essere chiamate dal codice. Pertanto, queste azioni API non sono incluse in AWS CLI and AWS SDKs.

AWS politiche gestite (predefinite) per AWS CodeBuild

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da AWS. Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni, in modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Le policy gestite forniscono CodeBuild anche le autorizzazioni per eseguire operazioni in altri servizi, come IAM, Amazon EC2, AWS CodeCommit, Amazon ECR, Amazon SNS e CloudWatch Amazon Events, come richiesto per le responsabilità degli utenti a cui è stata concessa la politica in questione. Ad esempio, la `AWSCodeBuildAdminAccess` policy è una policy utente di livello amministrativo che consente agli utenti con questa policy di creare e gestire le regole CloudWatch Events per le build di progetti e gli argomenti di Amazon SNS per le notifiche sugli eventi relativi al progetto (argomenti i cui nomi hanno il prefisso `arn:aws:codebuild:`), nonché di amministrare progetti e gruppi di report in CodeBuild. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche per AWS CodeBuild.

`AWSCodeBuildAdminAccess`

Fornisce l'accesso completo all'CodeBuild, inclusa l'autorizzazione per amministrare i progetti di CodeBuild.

`AWSCodeBuildDeveloperAccess`

Fornisce l'accesso CodeBuild ma non consente l'amministrazione dei progetti di compilazione.

`AWSCodeBuildReadOnlyAccess`

Fornisce accesso in sola lettura a CodeBuild.

Per accedere agli artefatti di output della build CodeBuild creati, è inoltre necessario allegare la policy gestita denominata `AWS::AmazonS3ReadOnlyAccess`.

Per creare e gestire i ruoli CodeBuild di servizio, è inoltre necessario allegare la policy AWS gestita denominata. `IAMFullAccess`

È inoltre possibile creare policy IAM personalizzate per concedere le autorizzazioni per operazioni e risorse CodeBuild. Puoi collegare queste policy personalizzate agli utenti o ai gruppi che richiedono le autorizzazioni.

Argomenti

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

AWSCodeBuildAdminAccess

La `AWSCodeBuildAdminAccess` policy fornisce l'accesso completo CodeBuild, incluse le autorizzazioni per amministrare i progetti di CodeBuild compilazione. Applica questa politica solo agli utenti di livello amministrativo per concedere loro il pieno controllo su CodeBuild progetti, gruppi di report e risorse correlate presenti nel tuo AWS account, inclusa la possibilità di eliminare progetti e gruppi di report.

Per la policy gestita completa, consulta [AWSCodeBuildAdminAccess](#) il riferimento alla policy gestita di IAM.

AWSCodeBuildDeveloperAccess

La `AWSCodeBuildDeveloperAccess` policy consente l'accesso a tutte le funzionalità delle risorse relative a progetti CodeBuild e gruppi di report. Questa politica non consente agli utenti di eliminare CodeBuild progetti o gruppi di report o risorse correlate in altri AWS servizi, come CloudWatch Eventi. È consigliabile applicare questa policy alla maggior parte degli utenti.

Per la policy gestita completa, consulta [AWSCodeBuildDeveloperAccess](#) il riferimento alla policy gestita di IAM.

AWSCodeBuildReadOnlyAccess

La `AWSCodeBuildReadOnlyAccess` policy garantisce l'accesso in sola lettura CodeBuild e alle relative risorse in altri servizi. AWS Applicare questo criterio agli utenti che possono visualizzare ed eseguire build, visualizzare progetti e visualizzare gruppi di report, ma non possono apportare modifiche.

Per la policy gestita completa, consulta [AWSCodeBuildReadOnlyAccess](#)il riferimento alla policy gestita di IAM.

CodeBuild politiche e notifiche gestite

CodeBuild supporta le notifiche, che possono notificare agli utenti modifiche importanti relative alla creazione di progetti. Le politiche gestite CodeBuild includono dichiarazioni politiche per la funzionalità di notifica. Per ulteriori informazioni, vedere [Cosa sono le notifiche?](#).

Autorizzazioni correlate alle notifiche nelle policy gestite di sola lettura

Le policy gestite `AWSCodeBuildReadOnlyAccess` includono le seguenti dichiarazioni per consentire l'accesso in sola lettura alle notifiche. Gli utenti con questa policy gestita applicata possono visualizzare le notifiche per le risorse, ma non possono crearle, gestirle o sottoscriverle.

```
{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition" : {
    "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}
```

Autorizzazioni correlate alle notifiche in altre policy gestite

La policy gestita `AWSCodeBuildDeveloperAccess` include le seguenti istruzioni per consentire agli utenti di creare, modificare e sottoscrivere le notifiche. Gli utenti non possono eliminare le regole di notifica o gestire i tag per le risorse.

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
```



```

    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}

```

Per ulteriori informazioni su IAM e le notifiche, consulta [Identity and Access ManagementAWS CodeStarNotifications](#).

CodeBuild aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CodeBuild da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS su. [AWS CodeBuild Cronologia dei documenti della Guida dell'utente](#)

Modifica	Descrizione	Data
AWSCodeBuildAdminAccess e AWSCodeBuildDeveloperAccess — Aggiornamento alle politiche esistenti	CodeBuild ha aggiunto l'<code>ssmmessages:OpenDataChannel</code> autorizzazione a queste politiche per supportare il debug interattivo delle build di Session Manager. Le <code>AWSCodeBuildDeveloperAccess</code> politiche <code>AWSCodeBuildAdminAccess</code> and ora includono l'<code>ssmmessages:OpenDataChannel</code> azione per le risorse di sessione di Session Manager (<code>arn:aws:ssm:*:*:session/*</code>) per supportare l'applicazione	1 dicembre 2025

Modifica	Descrizione	Data
	di SigV4 su questa API. WebSocket	
AWSCodeBuildAdminAccess , AWSCodeBuildDeveloperAccess , e AWSCodeBuildReadOnlyAccess — Aggiornamento alle politiche esistenti	CodeBuild ha aggiornato una risorsa su queste politiche. Le AWSCodeBuildReadOnlyAccess politiche AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess , e sono state modificate per aggiornare una risorsa esistente. La risorsa originale arn:aws:codebuild:* è stata aggiornata a arn:aws:codebuild:*:*:project/* .	15 novembre 2024
AWSCodeBuildAdminAccess ,AWSCodeBuildDeveloperAccess , e AWSCodeBuildReadOnlyAccess — Aggiornamento alle politiche esistenti	CodeBuild ha aggiunto una risorsa a queste politiche per supportare il AWS CodeConnections rebranding. Le AWSCodeBuildReadOnlyAccess politiche AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess , e sono state modificate per aggiungere una risorsa, arn:aws:codeconnections:*:*:*	18 aprile 2024

Modifica	Descrizione	Data
AWSCodeBuildAdminAccess e AWSCodeBuildDeveloperAccess — Aggiornamento alle politiche esistenti	CodeBuild ha aggiunto un'autorizzazione a queste politiche per supportare un tipo di notifica aggiuntivo o utilizzando Amazon Q Developer nelle applicazioni di chat. Le AWSCodeBuildDeveloperAccess politiche AWSCodeBuildAdminAccess e sono state modificate e per aggiungere un'autorizzazione, chatbot:ListMicrosoftTeamsChannelConfigurations .	16 maggio 2023
CodeBuild ha iniziato a tenere traccia delle modifiche	CodeBuild ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	16 maggio 2021

Esempi di policy gestite dal cliente

Questa sezione include policy utente di esempio che concedono autorizzazioni per operazioni AWS CodeBuild . Queste politiche funzionano quando si utilizza l' CodeBuild API AWS SDKs, o. AWS CLI. Quando si utilizza la console, devi concedere autorizzazioni specifiche aggiuntive alla console. Per informazioni, consulta [Autorizzazioni necessarie per utilizzare la console AWS CodeBuild](#).

Puoi utilizzare i seguenti esempi di policy IAM per limitare CodeBuild l'accesso a utenti e ruoli.

Argomenti

- [Concessione a un utente di ottenere informazioni sui progetti di compilazione](#)
- [Consenti a un utente di ottenere informazioni sulle flotte](#)
- [Concessione a un utente di ottenere informazioni sui gruppi di report](#)

- [Concessione a un utente di ottenere informazioni sui report](#)
- [Concessione a un utente di creare progetti di compilazione](#)
- [Consenti a un utente di creare una flotta](#)
- [Concessione a un utente di creare un gruppo di report](#)
- [Consenti a un utente di eliminare una flotta](#)
- [Concessione a un utente di eliminare un gruppo di report](#)
- [Concessione a un utente di eliminare un report](#)
- [Concessione a un utente di eliminare progetti di compilazione](#)
- [Concessione a un utente di ottenere un elenco di nomi di progetti di compilazione](#)
- [Concessione a un utente di modificare informazioni sui progetti di compilazione](#)
- [Consenti a un utente di modificare una flotta](#)
- [Concessione a un utente di modificare un gruppo di report](#)
- [Concessione a un utente di ottenere informazioni sulle compilazioni](#)
- [Consenti a un utente di ottenere un elenco di build IDs per un progetto di compilazione](#)
- [Consenti a un utente di ottenere un elenco di build IDs](#)
- [Consenti a un utente di ottenere un elenco di flotte](#)
- [Concessione a un utente di ottenere un elenco dei gruppi di report](#)
- [Concessione a un utente di ottenere un elenco dei report](#)
- [Concessione a un utente di ottenere un elenco dei report riferiti a un gruppo di report](#)
- [Concessione a un utente di ottenere un elenco dei casi di test riferiti a un report](#)
- [Concessione a un utente di avviare l'esecuzione di compilazioni](#)
- [Concessione a un utente di tentare di interrompere le compilazioni](#)
- [Concessione a un utente di tentare di eliminare le compilazioni](#)
- [Consenti a un utente di ottenere informazioni sulle immagini Docker gestite da CodeBuild](#)
- [Consenti a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service](#)
- [Consenti CodeBuild l'accesso ai AWS servizi necessari per creare un'interfaccia di rete VPC](#)
- [Utilizzate un'istruzione di negazione per impedire la disconnessione dai provider di origine AWS](#)
[CodeBuild](#)

Concessione a un utente di ottenere informazioni sui progetti di compilazione

La seguente istruzione di policy di esempio permette all'utente di ottenere informazioni sui progetti di compilazione, nella regione us-east-2 per l'account 123456789012, che iniziano con il nome my:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Consenti a un utente di ottenere informazioni sulle flotte

Il seguente esempio di dichiarazione politica consente a un utente di ottenere informazioni sulle flotte nella us-east-2 regione per conto: 123456789012

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetFleets",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Concessione a un utente di ottenere informazioni sui gruppi di report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere informazioni sui gruppi di report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Concessione a un utente di ottenere informazioni sui report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere informazioni sui report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Concessione a un utente di creare progetti di compilazione

La seguente dichiarazione politica di esempio consente a un utente di creare progetti di compilazione con qualsiasi nome, ma solo nella us-east-2 regione per account 123456789012 e utilizzando solo il ruolo di CodeBuild servizio specificato:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
    }
  ]
}
```

La seguente dichiarazione politica di esempio consente a un utente di creare progetti di compilazione con qualsiasi nome, ma solo nella us-east-2 regione per account 123456789012 e solo utilizzando il ruolo di CodeBuild servizio specificato. Implica inoltre che l'utente possa utilizzare il ruolo di servizio specificato solo con AWS CodeBuild e non altri AWS servizi.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
```

```
"Effect": "Allow",
"Action": "iam:PassRole",
"Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole",
"Condition": {
    "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
}
]
```

Consenti a un utente di creare una flotta

Il seguente esempio di dichiarazione politica consente a un utente di creare una flotta nella us-east-2 regione per l'account123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Concessione a un utente di creare un gruppo di report

La seguente istruzione di policy di esempio permette all'utente di creare un gruppo di report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```
    "Effect": "Allow",
    "Action": "codebuild:CreateReportGroup",
    "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
  }
]
```

Consenti a un utente di eliminare una flotta

Il seguente esempio di dichiarazione politica consente a un utente di eliminare una flotta nella us-east-2 regione per account123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Concessione a un utente di eliminare un gruppo di report

La seguente istruzione di policy esemplificativa permette all'utente di eliminare un gruppo di report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

```
}  
]  
}
```

Concessione a un utente di eliminare un report

La seguente istruzione di policy esemplificativa permette all'utente di eliminare un report nella regione us-east-2 per l'account 123456789012:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "codebuild:DeleteReport",  
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"  
    }  
  ]  
}
```

Concessione a un utente di eliminare progetti di compilazione

La seguente istruzione di policy esemplificativa permette all'utente di eliminare i progetti di compilazione nella regione us-east-2 per l'account 123456789012 che iniziano con il nome my:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "codebuild:DeleteProject",  
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"  
    }  
  ]  
}
```

Concessione a un utente di ottenere un elenco di nomi di progetti di compilazione

La seguente istruzione di policy di esempio permette all'utente di ottenere un elenco dei nomi di progetti di compilazione per lo stesso account.

JSON

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

Concessione a un utente di modificare informazioni sui progetti di compilazione

La seguente istruzione di policy esemplificativa permette all'utente di modificare informazioni sui progetti di compilazione con qualsiasi nome solo nella regione us-east-2 per l'account 123456789012 e di utilizzare unicamente il ruolo del servizio AWS CodeBuild specificato:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
    }
  ]
}
```

```
}
```

Consenti a un utente di modificare una flotta

Il seguente esempio di dichiarazione politica consente a un utente di modificare una flotta nella us-east-2 regione per account123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Concessione a un utente di modificare un gruppo di report

La seguente istruzione di policy esemplificativa permette all'utente di modificare un gruppo di report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Concessione a un utente di ottenere informazioni sulle compilazioni

La seguente istruzione di policy esemplificativa permette all'utente di ottenere informazioni sulla compilazione nella regione us-east-2 per l'account 123456789012, per i progetti di compilazione denominati my-build-project e my-other-build-project:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetBuilds",
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
      ]
    }
  ]
}
```

Consenti a un utente di ottenere un elenco di build IDs per un progetto di compilazione

La seguente dichiarazione politica di esempio consente a un utente di ottenere un elenco di build IDs nella us-east-2 regione per gli account 123456789012 dei progetti di compilazione denominati my-build-project e my-other-build-project:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
      ]
    }
  ]
}
```

```
}  
]  
}
```

Consenti a un utente di ottenere un elenco di build IDs

Il seguente esempio di dichiarazione politica consente a un utente di ottenere un elenco di tutte le build IDs per lo stesso account:

JSON

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

Consenti a un utente di ottenere un elenco di flotte

La seguente dichiarazione politica di esempio consente a un utente di ottenere un elenco di flotte nella us-east-2 regione per conto: 123456789012

JSON

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

Concessione a un utente di ottenere un elenco dei gruppi di report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere un elenco dei gruppi di report nella regione us-east-2 per l'account 123456789012:

JSON

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

Concessione a un utente di ottenere un elenco dei report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere un elenco dei report nella regione us-east-2 per l'account 123456789012:

JSON

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

Concessione a un utente di ottenere un elenco dei report riferiti a un gruppo di report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere un elenco dei report riferiti a un gruppo di report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportsForReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Concessione a un utente di ottenere un elenco dei casi di test riferiti a un report

La seguente istruzione di policy esemplificativa permette all'utente di ottenere un elenco dei casi di test riferiti a un report nella regione us-east-2 per l'account 123456789012:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DescribeTestCases",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```


Concessione a un utente di avviare l'esecuzione di compilazioni

La seguente istruzione di policy esemplificativa permette all'utente di eseguire compilazioni nella regione us-east-2 per l'account 123456789012, per qualsiasi progetto di compilazione che inizia con il nome my:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Concessione a un utente di tentare di interrompere le compilazioni

La seguente istruzione di policy di esempio permette a un utente di tentare di interrompere le compilazioni di esecuzione solo nella regione us-east-2 per l'account 123456789012 per qualsiasi progetto di compilazione che inizia con il nome my:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Concessione a un utente di tentare di eliminare le compilazioni

La seguente istruzione di policy di esempio permette all'utente di tentare di eliminare le compilazioni solo nella regione us-east-2 per l'account 123456789012 per qualsiasi progetto di compilazione che inizia con il nome my:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchDeleteBuilds",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Consenti a un utente di ottenere informazioni sulle immagini Docker gestite da CodeBuild

Il seguente un esempio di un'istruzione di policy che permette all'utente di ottenere informazioni su tutte le immagini Docker gestite da CodeBuild.

JSON

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

Consenti a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service

L'esempio seguente di dichiarazione sulla politica delle risorse consente a un utente di aggiungere una politica di autorizzazione VPC per un ruolo di fleet service:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVpcCreateNI",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
        "arn:aws:ec2:us-west-2:111122223333:security-group/security-
group-id-1",
        "arn:aws:ec2:us-west-2:111122223333:network-interface/*"
      ]
    },
    {
      "Sid": "CodeBuildFleetVpcPermission",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2>DeleteNetworkInterface"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeBuildFleetVpcNIPermission",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission"
      ],
    }
  ]
}
```

```

        "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
        "Condition": {
            "StringEquals": {
                "ec2:Subnet": [
                    "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1"
                ]
            }
        }
    ]
}

```

Il seguente esempio di dichiarazione sulla politica delle risorse consente a un utente di aggiungere una politica di autorizzazione Amazon Managed Image (AMI) personalizzata per un ruolo di fleet service:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeImages",
      "Resource": "*"
    }
  ]
}

```

L'esempio seguente di dichiarazione sulla politica di fiducia consente a un utente di aggiungere una politica di autorizzazione per un ruolo di fleet service:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVPCTrustPolicy",

```

```

    "Effect": "Allow",
    "Principal": {
      "Service": "codebuild.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
]
}

```

Consenti CodeBuild l'accesso ai AWS servizi necessari per creare un'interfaccia di rete VPC

La seguente dichiarazione politica di esempio concede AWS CodeBuild l'autorizzazione a creare un'interfaccia di rete in un VPC con due sottoreti:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2>DeleteNetworkInterface",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission"
      ],

```

```

    "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
    "Condition": {
      "StringEquals": {
        "ec2:AuthorizedService": "codebuild.amazonaws.com"
      },
      "ArnEquals": {
        "ec2:Subnet": [
          "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
          "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-2"
        ]
      }
    }
  }
]
}

```

Utilizzate un'istruzione di negazione per impedire la disconnessione dai provider di origine AWS CodeBuild

Il seguente esempio di istruzione della policy utilizza un'istruzione di rifiuto per impedire a AWS CodeBuild di disconnettersi dai provider di origine. Utilizza `codebuild:DeleteAuthToken`, che è l'inverso di `codebuild:PersistAuthToken` e `codebuild:ImportSourceCredentials`, per connettersi ai provider di origine. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per consentire alla AWS CodeBuild console di connettersi ai provider di origine](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "codebuild:DeleteAuthToken",
      "Resource": "*"
    }
  ]
}

```

AWS CodeBuild riferimento alle autorizzazioni

Puoi utilizzare i tasti AWS-wide condition nelle tue AWS CodeBuild politiche per esprimere condizioni. Per un elenco, consulta [Available Keys](#) nella IAM User Guide.

Puoi specificare le operazioni nel campo Action della policy. Per specificare un'operazione, utilizza il prefisso `codebuild:` seguito dal nome dell'operazione API (ad esempio, `codebuild:CreateProject` and `codebuild:StartBuild`). Per specificare più operazioni in una sola istruzione, separa ciascuna di esse con una virgola (ad esempio, "Action": [`"codebuild:CreateProject"`, `"codebuild:StartBuild"`]).

Utilizzo di caratteri jolly

Puoi specificare un ARN, con o senza un carattere jolly (*), come valore della risorsa nel campo Resource della policy. È possibile utilizzare un carattere jolly per specificare più operazioni o risorse. Ad esempio, `codebuild:*` specifica tutte le CodeBuild azioni e `codebuild:Batch*` specifica tutte le CodeBuild azioni che iniziano con la parola. Batch L'esempio seguente concede l'accesso completo a tutti i progetti di compilazione con nomi che iniziano con my:

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

CodeBuild Operazioni API e autorizzazioni richieste per le azioni

BatchDeleteBuilds

Operazione: `codebuild:BatchDeleteBuilds`

Necessario per eliminare compilazioni.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetBuilds

Operazione: `codebuild:BatchGetBuilds`

Necessario per ottenere informazioni sulle compilazioni.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetProjects

Operazione: `codebuild:BatchGetProjects`

Necessario per ottenere informazioni sui progetti di compilazione.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetReportGroups

Operazione: `codebuild:BatchGetReportGroups`

Necessario per ottenere informazioni sui gruppi di report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchGetReports

Operazione: `codebuild:BatchGetReports`

Necessario per ottenere informazioni sui report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchPutTestCases¹

Operazione: `codebuild:BatchPutTestCases`

Necessario per creare o aggiornare un report di test.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateProject

Operazioni: `codebuild>CreateProject`, `iam:PassRole`

Necessario per creare i progetti di compilazione.

Risorse:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

CreateReport¹

Operazione: `codebuild>CreateReport`

Obbligatorio per creare un report di test.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateReportGroup

Operazione: `codebuild:CreateReportGroup`

Necessario per creare un gruppo di report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateWebhook

Operazione: `codebuild:CreateWebhook`

Obbligatoria per creare un webhook.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteProject

Operazione: `codebuild>DeleteProject`

Necessario per eliminare un CodeBuild progetto.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteReport

Operazione: `codebuild>DeleteReport`

Necessario per eliminare un report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteReportGroup

Operazione: `codebuild>DeleteReportGroup`

Necessario per eliminare un gruppo di report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteSourceCredentials

Operazione: `codebuild>DeleteSourceCredentials`

Necessario per eliminare un set di `SourceCredentialsInfo` oggetti che contengono informazioni sulle credenziali per un GitHub repository GitHub Enterprise Server o Bitbucket.

Risorsa: *

DeleteWebhook

Operazione: `codebuild>DeleteWebhook`

Obbligatoria per creare un webhook.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DescribeTestCases

Operazione: `codebuild:DescribeTestCases`

Necessario per la restituzione di un elenco impaginato dei casi di test.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

ImportSourceCredentials

Operazione: `codebuild:ImportSourceCredentials`

Necessario per importare un set di `SourceCredentialsInfo` oggetti che contengono informazioni sulle credenziali per un repository GitHub Enterprise Server o GitHub Bitbucket.

Risorsa: *

InvalidateProjectCache

Operazione: `codebuild:InvalidateProjectCache`

Necessario per reimpostare la cache per un progetto.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuildBatches

Operazione: `codebuild:ListBuildBatches`

Necessario per ottenere un elenco dei batch di build. IDs

Risorsa: *

ListBuildBatchesForProject

Operazione: `codebuild:ListBuildBatchesForProject`

Necessario per ottenere un elenco dei batch di build IDs per un progetto specifico.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuilds

Operazione: `codebuild:ListBuilds`

Necessario per ottenere un elenco delle build IDs.

Risorsa: *

ListBuildsForProject

Operazione: `codebuild:ListBuildsForProject`

Necessario per ottenere un elenco delle build IDs per un progetto di compilazione.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListCuratedEnvironmentImages

Operazione: `codebuild:ListCuratedEnvironmentImages`

Necessario per permettere di ottenere informazioni sulle immagini Docker gestite da AWS CodeBuild.

Risorsa: * (necessario, ma non si riferisce a una risorsa AWS indirizzabile)

ListProjects

Operazione: `codebuild:ListProjects`

Necessario per ottenere un elenco di nomi di progetti di compilazione.

Risorsa: *

ListReportGroups

Operazione: `codebuild:ListReportGroups`

Necessario per ottenere un elenco dei gruppi di report.

Risorsa: *

ListReports

Operazione: `codebuild:ListReports`

Necessario per ottenere un elenco di report.

Risorsa: *

ListReportsForReportGroup

Operazione: `codebuild:ListReportsForReportGroup`

Necessario per ottenere un elenco dei report riferiti a un gruppo di report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

RetryBuild

Operazione: `codebuild:RetryBuild`

Necessario per riprovare le build.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StartBuild

Operazione: `codebuild:StartBuild`

Necessario per avviare progetti di compilazione.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StopBuild

Operazione: `codebuild:StopBuild`

Necessario per tentare di interrompere le compilazioni in esecuzione.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

UpdateProject

Operazioni: `codebuild:UpdateProject`, `iam:PassRole`

Necessario per modificare informazioni sulle compilazioni.

Risorse:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateProjectVisibility

Operazioni: `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Necessario per modificare la visibilità pubblica delle build di un progetto.

Risorse:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateReport¹

Operazione: `codebuild:UpdateReport`

Necessario per creare o aggiornare un report di test.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateReportGroup

Operazione: `codebuild:UpdateReportGroup`

Necessario per l'aggiornamento di un gruppo di report.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateWebhook

Operazione: `codebuild:UpdateWebhook`

Necessario per aggiornare un webhook.

Risorsa: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

¹ Utilizzato solo a scopo di autorizzazione. Non esiste alcuna API per questa azione.

Utilizzo dei tag per controllare l'accesso alle AWS CodeBuild risorse

Le condizioni nelle dichiarazioni politiche IAM fanno parte della sintassi che puoi utilizzare per specificare le autorizzazioni per le azioni basate sul CodeBuild progetto. Puoi creare una policy che consenta o neghi azioni sui progetti in base ai tag associati a tali progetti e quindi applicare tali policy ai gruppi IAM configurati per la gestione degli utenti. Per informazioni sull'applicazione di tag a un progetto utilizzando la console o AWS CLI, consulta [Crea un progetto di compilazione in AWS CodeBuild](#). Per informazioni sull'applicazione dei tag utilizzando l' CodeBuild SDK, consulta [CreateProject](#) e [Tag](#) nell'CodeBuildAPI Reference. Per informazioni sull'utilizzo dei tag per controllare l'accesso alle AWS risorse, consulta [Controlling Access to AWS Resources Using Resource Tags](#) nella IAM User Guide.

Important

Quando si utilizza la funzionalità di capacità riservata, i dati memorizzati nella cache delle istanze della flotta, inclusi i file sorgente, i layer Docker e le directory memorizzate nella cache specificate nelle specifiche di costruzione, possono essere accessibili ad altri progetti all'interno dello stesso account. Questa funzionalità è preimpostata e consente ai progetti all'interno dello stesso account di condividere le istanze del parco istanze.

Example Esempio 1: limita le azioni CodeBuild del progetto in base ai tag delle risorse

L'esempio seguente nega tutte le operazioni BatchGetProjects sui progetti contenenti tag con la chiave `Environment` e il valore della chiave `Production`. L'amministratore di un utente deve allegare questa policy IAM oltre alla policy utente gestita agli utenti non autorizzati. La chiave della condizione `aws:ResourceTag` viene utilizzata per controllare l'accesso alle risorse in base ai tag.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
```

```

        "codebuild:BatchGetProjects"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:ResourceTag/Environment": "Production"
        }
    }
}
]
}

```

Example Esempio 2: limita le azioni CodeBuild del progetto in base ai tag di richiesta

La policy seguente nega agli utenti l'autorizzazione all'operazione `CreateProject` se la richiesta contiene un tag con la chiave `Environment` e il valore della chiave `Production`. Inoltre, la policy impedisce a questi utenti non autorizzati di modificare i progetti utilizzando la chiave di condizione `aws:TagKeys` per non consentire `UpdateProject` se la richiesta contiene un tag con la chiave `Environment`. Un amministratore deve allegare questa policy IAM oltre alla policy utente gestita agli utenti che non sono autorizzati a eseguire queste azioni. La chiave `aws:RequestTag condition` viene utilizzata per controllare quali tag possono essere passati in una richiesta IAM

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:CreateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/Environment": "Production"
        }
      }
    },
    {
      "Effect": "Deny",

```

```
"Action": [
  "codebuild:UpdateProject"
],
"Resource": "*",
"Condition": {
  "ForAnyValue:StringEquals": {
    "aws:TagKeys": ["Environment"]
  }
}
}
```

Example Esempio 3: negare o consentire azioni sui gruppi di report in base ai tag delle risorse

Puoi creare una policy che consenta o neghi azioni sulle CodeBuild risorse (progetti e gruppi di report) in base ai AWS tag associati a tali risorse e quindi applicare tali policy ai gruppi IAM configurati per la gestione degli utenti. Ad esempio, puoi creare una policy che neghi tutte le CodeBuild azioni su qualsiasi gruppo di report con la chiave AWS tag Status e il valore chiave diSecret, e quindi applicare tale policy al gruppo IAM che hai creato per sviluppatori generici (*Developers*). Devi quindi assicurarti che gli sviluppatori che lavorano su quei gruppi di report con tag non siano membri di quel *Developers* gruppo generale, ma appartengano invece a un gruppo IAM diverso a cui non è applicata la politica restrittiva (*SecretDevelopers*).

L'esempio seguente nega tutte le CodeBuild azioni sui gruppi di report contrassegnati con la chiave Status e il valore chiave di: Secret

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild>DeleteReportGroup",
        "codebuild>ListReportGroups",
```



```

        "codebuild:ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:RequestedRegion": "us-east-1"
        }
    }
}
]
}

```

Example Esempio 4: limita CodeBuild le azioni in AWSCode BuildDeveloperAccess base ai tag delle risorse

Puoi creare politiche che consentano CodeBuild azioni su tutti i gruppi di report e i progetti che non sono etichettati con tag specifici. Ad esempio, la policy seguente consente l'equivalente delle autorizzazioni [AWSCodeBuildDeveloperAccess](#) per tutti i gruppi di report e progetti tranne quelli contrassegnati con i tag specificati:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",

```

```
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*",
    "Condition": {
        "StringNotEquals": {
            "aws:ResourceTag/Status": "Secret",
            "aws:ResourceTag/Team": "Saanvi"
        }
    }
}
```

Visualizzazione di risorse nella console

La AWS CodeBuild console richiede l'`ListRepositories` autorizzazione per visualizzare un elenco di repository per il tuo AWS account nella AWS regione in cui hai effettuato l'accesso. La console, inoltre, include a funzione `Go to resource` (Vai alla risorsa) per eseguire rapidamente una ricerca di risorse senza distinzione tra maiuscole e minuscole. Questa ricerca viene eseguita nel tuo AWS account nella AWS regione in cui hai effettuato l'accesso. Le seguenti risorse sono visualizzate per i seguenti servizi:

- AWS CodeBuild: progetti di compilazione
- AWS CodeCommit: repository
- AWS CodeDeploy: applicazioni
- AWS CodePipeline: pipeline

Per eseguire la ricerca nelle risorse di tutti i servizi, è necessario disporre delle autorizzazioni seguenti:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

I risultati non vengono restituiti per le risorse di un servizio se non hai le autorizzazioni per quel servizio. Anche se hai le autorizzazioni per la visualizzazione delle risorse, alcune risorse non verranno restituite se c'è un esplicito Deny a visualizzare tali risorse.

Convalida della conformità per AWS CodeBuild

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la [Documentazione AWS sulla sicurezza](#).

Resilienza in AWS CodeBuild

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

[Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura globale.AWS](#)

Sicurezza dell'infrastruttura in AWS CodeBuild

In quanto servizio gestito, AWS CodeBuild è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere CodeBuild attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Accedi al tuo provider di origine in CodeBuild

Per il GitHub nostro GitHub Enterprise Server, si utilizza un token di accesso personale, un segreto di Secrets Manager, una connessione o un' OAuth app per accedere al provider di origine. Per Bitbucket, utilizzi un token di accesso, una password dell'app, un segreto di Secrets Manager, una connessione o un' OAuth app per accedere al provider di origine.

Argomenti

- [Crea e archivia un token in un segreto di Secrets Manager](#)
- [GitHub e l'accesso a GitHub Enterprise Server in CodeBuild](#)
- [Accesso a Bitbucket in CodeBuild](#)
- [GitLab accesso in CodeBuild](#)

Crea e archivia un token in un segreto di Secrets Manager

Se si sceglie di archiviare il token di accesso utilizzando Secrets Manager, è possibile utilizzare una connessione segreta esistente o creare un nuovo segreto. Per creare un nuovo segreto, procedi come segue:

Console di gestione AWS

Per creare un segreto di Secrets Manager nel Console di gestione AWS

1. Come provider di origine, scegli Bitbucket o GitHubGitHub Enterprise.
2. Per Credential, esegui una delle seguenti operazioni:
 - Scegli Credenziale di origine predefinita per utilizzare la credenziale di origine predefinita del tuo account da applicare a tutti i progetti.
 - a. Se non sei connesso al tuo provider di origine, scegli Gestisci credenziali di origine predefinite.
 - b. Per Tipo di credenziale, scegli un tipo di credenziale diverso da. CodeConnections

- c. Per Service, scegli Secrets Manager e per Secrets scegli New secret.
 - d. In Nome segreto, inserisci il nome del tuo segreto.
 - e. In Descrizione segreta, facoltativa, inserisci una descrizione per il tuo segreto.
 - f. A seconda del provider di origine che hai scelto, inserisci il token o il nome utente e la password dell'app e scegli Salva.
- Scegli Credenziale di origine personalizzata per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni predefinite del tuo account.
 - a. Per Tipo di credenziale, scegli un tipo di credenziale diverso da. CodeConnections
 - b. In Connessione, scegli Crea un segreto.
 - c. In Nome segreto, inserisci il nome del tuo segreto.
 - d. In Descrizione segreta, facoltativa, inserisci una descrizione per il tuo segreto.
 - e. A seconda del provider di origine che hai scelto, inserisci il token o il nome utente e la password dell'app e scegli Crea.

AWS CLI

Per creare un segreto di Secrets Manager nel AWS CLI

- Apri un terminale (Linux, macOS o Unix) o prompt dei comandi (Windows). Utilizzare il AWS CLI per eseguire il create-secret comando Secrets Manager.

```
aws secretsmanager create-secret --region <aws-region> \
    --name '<secret-name>' \
    --description '<secret-description>' \
    --secret-string '{
        "ServerType": "<server-type>",
        "AuthType": "<auth-type>",
        "Token": "<token>"
    }' \
    --tags Key=codebuild:source,Value='' \
        Key=codebuild:source:type,Value=<type> \
        Key=codebuild:source:provider,Value=<provider>
```

I segreti di Secrets Manager CodeBuild accettati devono trovarsi nello stesso account e AWS nella stessa regione del CodeBuild progetto e devono essere nel seguente formato JSON:

```
{
    "ServerType": ServerType,
    "AuthType": AuthType,
    "Token": string,
    "Username": string // Optional and is only used for Bitbucket app
    password
}
```

Campo	Valori validi	Description
ServerType	GITHUB GITHUB_ENTERPRISE BITBUCKET	Il fornitore di sorgenti di terze parti per il tuo segreto di Secrets Manager.
AuthType	PERSONAL_ACCESS_TO_KEN BASIC_AUTH	Il tipo di token di accesso utilizzato dalle credenziali. Perché GitHub, solo PERSONAL_ACCESS_TO_KEN è valido. BASIC_AUTH è valido solo per la password dell'app Bitbucket.
Token	<i>string</i>	Per la nostra GitHub GitHub azienda, questo è il token di accesso personale. Per Bitbucket, si tratta del token di accesso o della password dell'app Bitbucket.

Campo	Valori validi	Description
Username	<i>string</i>	Il nome utente di Bitbucket quando è BASIC_AUTH. AuthType Questo parametro non è valido per altri tipi di provider di sorgenti.

Inoltre, CodeBuild utilizza i seguenti tag di risorsa sul segreto per garantire che i segreti siano facilmente selezionabili durante la creazione o la modifica di progetti.

Chiave tag	Valore di tag	Description
codebuild:source:provider	github	Indica a CodeBuild quale provider è destinato questo segreto.
	github_enterprise	
	bitbucket	
codebuild:source:type	personal_access_token	Indica CodeBuild il tipo di token di accesso contenuto in questo segreto.
	basic_auth	

GitHub e l'accesso a GitHub Enterprise Server in CodeBuild

Infatti GitHub, puoi utilizzare un token di accesso personale, un' OAuth app, un segreto di Secrets Manager o una connessione all' GitHub app per accedere al provider di origine. Per GitHub Enterprise Server, è possibile utilizzare un token di accesso personale, un segreto di Secrets Manager o una connessione GitHub App per accedere al provider di origine.

Argomenti

- [GitHub Connessioni all'app per GitHub GitHub Enterprise Server](#)
- [GitHub e token di accesso a GitHub Enterprise Server](#)
- [GitHub OAuth app](#)

GitHub Connessioni all'app per GitHub GitHub Enterprise Server

Puoi usare l' GitHub app per connetterti con CodeBuild. GitHub Le connessioni alle app sono supportate tramite [AWS CodeConnections](#).

L'accesso al provider di origine consente di attivare una build [CreateWebhook](#) sottoscrivendo o [GitHub eventi webhook](#) utilizzando [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#) in CodeBuild.

Note

CodeConnections è disponibile in un numero di regioni inferiore a. CodeBuild È possibile utilizzare connessioni interregionali in CodeBuild. Le connessioni create nelle regioni opzionali non possono essere utilizzate in altre regioni. Per ulteriori informazioni, consulta [Endpoint e quote per AWS CodeConnections](#).

Argomenti

- [Passaggio 1: creare una connessione all' GitHub app \(console\)](#)
- [Fase 2: concedere al CodeBuild progetto l'accesso al ruolo IAM per utilizzare la connessione](#)
- [Fase 3: Configurazione CodeBuild per l'utilizzo della nuova connessione](#)
- [Risoluzione dei problemi con l'app GitHub](#)

Passaggio 1: creare una connessione all' GitHub app (console)

Segui questi passaggi per utilizzare la CodeBuild console in cui aggiungere una connessione per il tuo progetto GitHub.

Per creare una connessione a GitHub

- Segui le istruzioni nella Guida per l'utente degli strumenti per sviluppatori per [creare una connessione a GitHub](#).

 Note

Invece di creare o utilizzare una connessione esistente nel tuo account, puoi utilizzare una connessione condivisa da un altro AWS account. Per ulteriori informazioni, consulta [Condividere le connessioni con AWS gli account](#).

Fase 2: concedere al CodeBuild progetto l'accesso al ruolo IAM per utilizzare la connessione

Puoi concedere l'accesso al ruolo IAM del CodeBuild progetto per utilizzare GitHub i token forniti dalla tua connessione.

Per concedere l'accesso al CodeBuild ruolo IAM del progetto

1. Crea un ruolo IAM per il tuo CodeBuild progetto seguendo le istruzioni [Consenti CodeBuild di interagire con altri servizi AWS](#) relative al tuo CodeBuild progetto.
2. Seguendo le istruzioni, aggiungi la seguente politica IAM al ruolo CodeBuild del tuo progetto per concedere l'accesso alla connessione.

JSON

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

Fase 3: Configurazione CodeBuild per l'utilizzo della nuova connessione

È possibile configurare una connessione come credenziale a livello di account e utilizzarla in un progetto.

Console di gestione AWS

Per configurare una connessione come credenziale a livello di account in Console di gestione AWS

1. Per Provider di origine, scegli GitHub.
2. Per Credential, effettuate una delle seguenti operazioni:
 - Scegli Credenziale di origine predefinita per utilizzare la credenziale di origine predefinita del tuo account da applicare a tutti i progetti.
 - a. Se non sei connesso a GitHub, scegli Gestisci credenziali di origine predefinite.
 - b. Per il tipo di credenziale, scegli GitHub App.
 - c. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.
 - Scegli Credenziale di origine personalizzata per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni predefinite del tuo account.
 - a. Per Tipo di credenziale, scegli App. GitHub
 - b. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.

AWS CLI

Per configurare una connessione come credenziale a livello di account in AWS CLI

- Apri un terminale (Linux, macOS o Unix) o prompt dei comandi (Windows). Utilizzate il AWS CLI per eseguire il `import-source-credentials` comando, specificando il `--auth-type--` `server-type`, e `--token` per la connessione.

Utilizza il seguente comando:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type GITHUB --token <connection-arn>
```

Puoi anche impostare più token per i tuoi CodeBuild progetti. Per ulteriori informazioni, consulta [Configura più token come credenziali a livello di origine](#).

Risoluzione dei problemi con l'app GitHub

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi all' GitHub App.

Argomenti

- [Installa il AWS Connector per GitHub l'app in una regione indesiderata](#)
- [La connessione all' GitHub app non ha accesso agli archivi](#)
- [Al ruolo IAM del AWS servizio mancano le autorizzazioni IAM necessarie.](#)

Installa il AWS Connector per GitHub l'app in una regione indesiderata

Problema: hai installato il AWS Connector for GitHub dal GitHub Marketplace, ma la connessione è stata creata in un'area indesiderata. Se tenti di riconfigurare l'app sul GitHub sito Web, non funzionerà perché l'app è già installata sul tuo GitHub account.

Possibile causa: l'app è già installata nel tuo GitHub account, quindi puoi solo riconfigurare le autorizzazioni dell'app.

Soluzione consigliata: è possibile creare una nuova connessione con l'ID di installazione nella regione desiderata.

1. Apri la CodeConnections console in <https://console.aws.amazon.com/codesuite/impostazioni/connessioni> e accedi alla regione desiderata utilizzando il selettore di regione nella barra di navigazione della AWS console.
2. Segui le istruzioni nella Guida per l'utente degli strumenti per sviluppatori per [creare una connessione a. GitHub](#)

Note

Poiché hai già installato l' GitHub app AWS Connector for, puoi sceglierlo invece di installare una nuova app.

La connessione all' GitHub app non ha accesso agli archivi

Problema: un AWS servizio che utilizza la connessione, ad esempio CodeBuild o CodePipeline, segnala di non avere accesso al repository o che il repository non esiste. Alcuni possibili messaggi di errore includono:

- Authentication required for primary source.
- Unable to create webhook at this time. Please try again later.
- Failed to create webhook. GitHub API limit reached. Please try again later.

Possibile causa: potresti aver utilizzato l' GitHub app e non aver concesso l'ambito di autorizzazione del webhook.

Soluzione consigliata: per concedere l'ambito di autorizzazione richiesto, segui le istruzioni in [Navigazione verso l' GitHub app che desideri esaminare o modificare](#) per configurare l'app installata. Nella sezione Autorizzazioni, vedrai che l'app non dispone dell'autorizzazione per i webhook e che c'è un'opzione per rivedere le nuove autorizzazioni richieste. Rivedi e accetta le nuove autorizzazioni. Per ulteriori informazioni, consulta [Approvazione delle autorizzazioni aggiornate per un'app](#). GitHub

Possibile causa: la connessione funzionava come previsto, ma all'improvviso non ha accesso ai repository.

Soluzione possibile: inizia esaminando [le autorizzazioni e le installazioni](#), quindi verifica che l' GitHub App sia autorizzata e installata. Se l'installazione dell' GitHub App è sospesa, è necessario annullarla. Se l' GitHub App non è autorizzata per una connessione [UAT \(User Access Token\)](#) o non è installata per una connessione [IAT \(Installation Access Token\)](#), la connessione esistente non è più utilizzabile e sarà necessario creare una nuova connessione. Tieni presente che la reinstallazione dell' GitHub app non ripristinerà la connessione precedente associata alla vecchia installazione.

Soluzione possibile: se la connessione è una connessione UAT, assicurati che la connessione non venga utilizzata contemporaneamente, ad esempio in più esecuzioni CodeBuild simultanee di build. Questo perché invalida GitHub immediatamente un UAT emesso in precedenza se un token in scadenza viene aggiornato dalla connessione. Se è necessario utilizzare la connessione UAT per più CodeBuild build simultanee, è possibile creare più connessioni e utilizzare ciascuna connessione in modo indipendente.

Soluzione possibile: se la connessione UAT non è stata utilizzata negli ultimi 6 mesi, la connessione verrà invalidata da GitHub. Per risolvere questo problema, crea una nuova connessione.

Possibile causa: potresti aver utilizzato una connessione UAT senza installare l'app.

Soluzione consigliata: sebbene la creazione di una connessione UAT non richieda l'associazione della connessione a un'installazione GitHub dell'app, è necessaria un'installazione per rendere accessibile l'archivio. Segui le istruzioni per [rivedere le installazioni](#) e assicurarti che l' GitHub app sia installata. Se non è installata, vai alla [pagina dell'GitHub app](#) per installarla. Per ulteriori informazioni sull'accesso di UAT, consulta [Informazioni sui token di accesso degli utenti](#).

Al ruolo IAM del AWS servizio mancano le autorizzazioni IAM necessarie.

Problema: viene visualizzato uno dei seguenti messaggi di errore:

- Access denied to connection `<connection-arn>`
- Failed to get access token from `<connection-arn>`

Soluzione consigliata: in genere si utilizza una connessione con un AWS servizio, ad esempio CodePipeline o CodeBuild. Quando assegnate al AWS servizio un ruolo IAM, il AWS servizio può utilizzare l'autorizzazione del ruolo per agire per vostro conto. Assicurati che il ruolo IAM disponga delle autorizzazioni necessarie. Per ulteriori informazioni sull'autorizzazione IAM necessaria, consulta [Concedere al CodeBuild progetto l'accesso al ruolo IAM per utilizzare la connessione e la gestione delle identità e degli accessi per AWS CodeStar le notifiche e CodeConnections](#) nella Guida per l'utente della console Developer Tools.

GitHub e token di accesso a GitHub Enterprise Server

Prerequisiti del token di accesso

Prima di iniziare, è necessario aggiungere gli ambiti di autorizzazione appropriati al token di GitHub accesso.

Infatti GitHub, il token di accesso personale deve avere i seguenti ambiti.

- repo: concede il controllo completo dei repository privati.
- repo:status: concede l' read/write accesso agli stati di commit dei repository pubblici e privati.

- `admin:repo_hook`: concede il controllo completo degli hook di repository. Questo ambito non è necessario se il token ha l'ambito `repo`.
- `admin:org_hook`: Garantisce il pieno controllo degli hook organizzativi. Questo ambito è necessario solo se si utilizza la funzionalità webhook dell'organizzazione.

Per ulteriori informazioni, consulta [Comprendere gli ambiti OAuth delle app](#) sul GitHub sito Web.

Se utilizzi token di accesso personali dettagliati, a seconda del caso d'uso, il token di accesso personale potrebbe richiedere le seguenti autorizzazioni:

- **Contenuto: Sola lettura**: concede l'accesso agli archivi privati. Questa autorizzazione è necessaria se si utilizzano repository privati come fonte.
- **Stati di commit: Lettura e scrittura**: concede l'autorizzazione a creare stati di commit. Questa autorizzazione è necessaria se nel progetto è configurato un webhook o se è abilitata la funzione di report sullo stato della compilazione.
- **Webhook: Lettura e scrittura**: concede l'autorizzazione a gestire i webhook. Questa autorizzazione è necessaria se nel progetto è configurato un webhook.
- **Richieste pull: Sola lettura**: concede l'autorizzazione ad accedere alle richieste pull. Questa autorizzazione è necessaria se il webhook ha un `FILE_PATH` filtro sugli eventi di pull request.
- **Amministrazione: Lettura e scrittura**: questa autorizzazione è richiesta se si utilizza la funzionalità self-hosted GitHub Actions runner con CodeBuild. Per maggiori dettagli, consulta [Creare un token di registrazione per un repository](#) e [Tutorial: configura un GitHub Actions CodeBuild runner ospitato](#).

Note

Se desideri accedere agli archivi dell'organizzazione, assicurati di specificare l'organizzazione come proprietaria della risorsa del token di accesso.

Per ulteriori informazioni, consulta [Autorizzazioni richieste per i token di accesso personali dettagliati](#) sul sito Web. GitHub

Connect GitHub con un token di accesso (console)

Per utilizzare la console per connettere il progetto all' GitHub utilizzo di un token di accesso, procedi come segue quando crei un progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

1. Per Provider di origine, scegli GitHub.
2. Per Credential, effettuate una delle seguenti operazioni:
 - Scegli di utilizzare le credenziali dell'account per applicare le credenziali di origine predefinite dell'account a tutti i progetti.
 - a. Se non sei connesso a GitHub, scegli Gestisci le credenziali dell'account.
 - b. Per il tipo di credenziale, scegli Token di accesso personale.
 - Se hai scelto di utilizzare credenziali a livello di account per il Servizio, scegli quale servizio desideri utilizzare per archiviare il token e procedi come segue:
 - a. Se scegli di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova, quindi scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
 - b. Se scegli di utilizzarlo CodeBuild, inserisci il tuo token di accesso GitHub personale, quindi scegli Salva.
 - Seleziona Usa le credenziali di sostituzione solo per questo progetto per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni delle credenziali del tuo account.
 - a. Dall'elenco di credenziali compilato, scegli una delle opzioni in Token di accesso personale.
 - b. Puoi anche creare un nuovo token di accesso personale selezionando Crea una nuova connessione con token di accesso personale nella descrizione.

Connect GitHub con un token di accesso (CLI)

Segui questi passaggi per collegare il tuo progetto AWS CLI all' GitHub utilizzo di un token di accesso. Per informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta [Guida di riferimento alla riga di comando](#).

1. Eseguire il comando `import-source-credentials`:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Nell'output vengono visualizzati dati in formato JSON. Copiare i dati in un file (ad esempio, *import-source-credentials.json*) in una posizione del computer o dell'istanza locale in cui AWS CLI è installato. Modificare i dati copiati come segue, quindi salvare i risultati.

```
{
  "serverType": "server-type",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Sostituisci quanto segue:

- *server-type*: Valore richiesto. Il provider di origine utilizzato per queste credenziali. I valori validi sono GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB e GITLAB_SELF_MANAGED.
- *auth-type*: Valore richiesto. Il tipo di autenticazione utilizzato per connettersi a un repository. I valori validi sono OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS e SECRETS_MANAGER. Perché, è consentito solo GitHub PERSONAL_ACCESS_TOKEN. BASIC_AUTH è consentito solo con la password dell'app Bitbucket.
- *should-overwrite*: Valore opzionale. Imposta su `false` per evitare di sovrascrivere le credenziali di origine del repository. Imposta su `true` per sovrascrivere le credenziali di origine del repository. Il valore predefinito è `true`.
- *token*: valore obbligatorio. Per il GitHub nostro GitHub Enterprise Server, questo è il token di accesso personale. Per Bitbucket, si tratta del token di accesso personale o della password dell'app. Per il tipo di autenticazione CODECONNECTIONS, questo è l'ARN della connessione. Per il tipo di autenticazione SECRETS_MANAGER, questo è l'ARN segreto.
- *username*: Valore opzionale. Questo parametro viene ignorato per i provider GitHub di origine di GitHub Enterprise Server.

2. Per collegare l'account con un token di accesso, passare alla directory che contiene il file `import-source-credentials.json` salvato nel passaggio 1 ed eseguire nuovamente il comando `import-source-credentials`.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

I dati in formato JSON sono presenti nell'output con un Amazon Resource Name (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Se si esegue il comando `import-source-credentials` con lo stesso `server-type` e `auth-type` una seconda volta, il token di accesso archiviato viene aggiornato.

Dopo aver collegato l'account con un token di accesso, è possibile utilizzarlo `create-project` per creare il CodeBuild progetto. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).

3. Per visualizzare i token di accesso connessi, eseguire il comando `list-source-credentials`.

```
aws codebuild list-source-credentials
```

Un oggetto `sourceCredentialsInfos` in formato JSON è presente nell'output:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "server-type",
      "arn": "arn"
    }
  ]
}
```

L'elemento `sourceCredentialsObject` contiene un elenco di informazioni sulle credenziali di origine connesse:

- L'elemento `authType` è il tipo di autenticazione utilizzato dalle credenziali. Può essere `OAUTH`, `BASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS`, o `SECRETS_MANAGER`.
 - L'elemento `serverType` è il tipo di provider di origine. Questo può essere `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB`, o `GITLAB_SELF_MANAGED`.
 - L'elemento `arn` è l'ARN del token.
4. Per disconnettersi da un provider di origine e rimuovere i token di accesso, eseguire il comando `delete-source-credentials` con l'ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Vengono restituiti dati in formato JSON con l'ARN delle credenziali eliminate.

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

GitHub OAuth app

Connect GitHub tramite OAuth (console)

Per utilizzare la console per connettere il progetto all' GitHub utilizzo di un' OAuth app, esegui le seguenti operazioni quando crei un progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

1. Per Provider di origine, scegli GitHub.
2. Per Credential, effettuate una delle seguenti operazioni:
 - Scegli di utilizzare le credenziali dell'account per applicare le credenziali di origine predefinite dell'account a tutti i progetti.
 - a. Se non sei connesso a GitHub, scegli Gestisci le credenziali dell'account.
 - b. Per il tipo di credenziale, scegli OAuth l'app.

- Se hai scelto di utilizzare le credenziali a livello di account per il Servizio, scegli quale servizio desideri utilizzare per archiviare il token e procedi come segue:
 - a. Se scegli di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova, quindi scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
 - b. Se scegli di utilizzare, CodeBuild quindi scegli Salva.
- Seleziona Usa le credenziali di sostituzione solo per questo progetto per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni delle credenziali del tuo account.
 - a. Dall'elenco di credenziali compilato, scegli una delle opzioni in app. OAuth
 - b. Puoi anche creare un nuovo token per OAuth l'app selezionando Crea una nuova connessione con il token dell'app OAuth nella descrizione.

Per esaminare le OAuth app autorizzate, accedi a [Applicazioni](#) su GitHub e verifica che sia presente nell'elenco un'applicazione denominata AWS CodeBuild (*region*) di proprietà di [aws-codesuite](#).

Accesso a Bitbucket in CodeBuild

Per Bitbucket, utilizzi un token di accesso, una password per l'app, un'app o una connessione Bitbucket per accedere al provider di origine. OAuth

Argomenti

- [Connessioni all'app Bitbucket](#)
- [Password o token di accesso dell'app Bitbucket](#)
- [App Bitbucket OAuth](#)

Connessioni all'app Bitbucket

Puoi usare Bitbucket per connetterti con. CodeBuild Le connessioni all'app Bitbucket sono supportate tramite. [AWS CodeConnections](#)

 Note

CodeConnections è disponibile in meno aree geografiche di CodeBuild. È possibile utilizzare connessioni interregionali in CodeBuild. Le connessioni create nelle regioni opzionali non possono essere utilizzate in altre regioni. Per ulteriori informazioni, consulta [Endpoint e quote per AWS CodeConnections](#).

Argomenti

- [Passaggio 1: crea una connessione a Bitbucket \(console\)](#)
- [Fase 2: concedere al CodeBuild progetto l'accesso al ruolo IAM per utilizzare la connessione](#)
- [Fase 3: Configurazione CodeBuild per l'utilizzo della nuova connessione](#)

Passaggio 1: crea una connessione a Bitbucket (console)

Utilizza questi passaggi per utilizzare la CodeBuild console per aggiungere una connessione per il tuo progetto in Bitbucket.

Per creare una connessione a Bitbucket

- Segui le istruzioni nella Guida per l'utente di Developer Tools per [creare una connessione a Bitbucket](#).

 Note

Invece di creare o utilizzare una connessione esistente nel tuo account, puoi utilizzare una connessione condivisa da un altro AWS account. Per ulteriori informazioni, consulta [Condividere le connessioni con AWS gli account](#).

Fase 2: concedere al CodeBuild progetto l'accesso al ruolo IAM per utilizzare la connessione

Puoi concedere l'accesso al ruolo IAM del CodeBuild progetto per utilizzare i token Bitbucket forniti dalla tua connessione.

Per concedere CodeBuild l'accesso al ruolo IAM del progetto

1. Crea un ruolo IAM per il tuo CodeBuild progetto seguendo le istruzioni [Consenti CodeBuild di interagire con altri servizi AWS](#) relative al tuo CodeBuild progetto.
2. Seguendo le istruzioni, aggiungi la seguente politica IAM al ruolo CodeBuild del tuo progetto per concedere l'accesso alla connessione.

JSON

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

Fase 3: Configurazione CodeBuild per l'utilizzo della nuova connessione

È possibile configurare una connessione come credenziale a livello di account e utilizzarla in un progetto.

Console di gestione AWS

Per configurare una connessione come credenziale a livello di account in Console di gestione AWS

1. In Source provider (Provider origine) selezionare Bitbucket.
2. Per Credential, esegui una delle seguenti operazioni:
 - Scegli Credenziale di origine predefinita per utilizzare la credenziale di origine predefinita del tuo account da applicare a tutti i progetti.

- a. Se non sei connesso a Bitbucket, scegli Gestisci credenziali di origine predefinite.
- b. Per Tipo di credenziale, scegli. CodeConnections
- c. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.
- Scegli Credenziale di origine personalizzata per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni predefinite del tuo account.
 - a. Per Tipo di credenziale, scegli. CodeConnections
 - b. In Connessione, scegli di utilizzare una connessione esistente o crearne una nuova.

AWS CLI

Per configurare una connessione come credenziale a livello di account in AWS CLI

- Apri un terminale (Linux, macOS o Unix) o prompt dei comandi (Windows). Utilizzate il AWS CLI per eseguire il `import-source-credentials` comando, specificando il `--auth-type--server-type`, e `--token` per la connessione.

Utilizza il seguente comando:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type BITBUCKET --token <connection-arn>
```

Per ulteriori informazioni sulla configurazione di più token nel CodeBuild progetto, consulta. [Configura più token come credenziali a livello di origine](#)

Password o token di accesso dell'app Bitbucket

Prerequisiti

Prima di iniziare, devi aggiungere gli ambiti di autorizzazione appropriati alla password o al token di accesso dell'app Bitbucket.

Per Bitbucket, la password o il token di accesso dell'app devono avere i seguenti ambiti.

- `repository:read`: concede l'accesso in lettura a tutti i repository a cui ha accesso l'utente che fornisce l'autorizzazione.

- `pullrequest:read`: concede l'accesso in lettura alle richieste di pull. Se il tuo progetto ha un webhook Bitbucket, la password o il token di accesso dell'app devono avere questo ambito.
- `webhook`: concede l'accesso ai webhook. Se il progetto prevede un'operazione webhook, la password o il token di accesso dell'app devono avere questo ambito.
- `account`: consente l'accesso in lettura alle informazioni dell'account dell'utente.

Per ulteriori informazioni, consulta l'[API REST di Scopes for Bitbucket Cloud](#) e [OAuth su Bitbucket Cloud sul sito Web di Bitbucket](#).

Connect Bitbucket con una password per l'app (console)

Per utilizzare la console per connettere il tuo progetto a Bitbucket utilizzando una password per l'app, esegui le seguenti operazioni quando crei un progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

1. In Source provider (Provider origine) selezionare Bitbucket.
2. Per Credential, esegui una delle seguenti operazioni:
 - Scegli di utilizzare le credenziali dell'account per applicare le credenziali di origine predefinite dell'account a tutti i progetti.
 - a. Se non sei connesso a Bitbucket, scegli Gestisci le credenziali dell'account.
 - b. Per il tipo di credenziale, scegli la password dell'app.
 - Se hai scelto di utilizzare le credenziali a livello di account per il Servizio, scegli quale servizio desideri utilizzare per archiviare il token e procedi come segue:
 - a. Se scegli di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova, quindi scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
 - b. Se scegli di utilizzarlo CodeBuild, inserisci il nome utente e la password dell'app Bitbucket, quindi scegli Salva.
 - Seleziona Usa le credenziali di sostituzione solo per questo progetto per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni delle credenziali del tuo account.
 - a. Dall'elenco di credenziali compilato, scegli una delle opzioni in Password dell'app.

- b. Puoi anche creare un nuovo token per la password dell'app selezionando Crea una nuova connessione con password per l'app nella descrizione.

Connect Bitbucket con un token di accesso (console)

Per utilizzare la console per connettere il tuo progetto a Bitbucket utilizzando un token di accesso, esegui le seguenti operazioni quando crei un progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

1. In Source provider (Provider origine) selezionare Bitbucket.
2. Per Credential, esegui una delle seguenti operazioni:
 - Scegli di utilizzare le credenziali dell'account per applicare le credenziali di origine predefinite dell'account a tutti i progetti.
 - a. Se non sei connesso a Bitbucket, scegli Gestisci le credenziali dell'account.
 - b. Per il tipo di credenziale, scegli Token di accesso personale.
 - Se hai scelto di utilizzare credenziali a livello di account per il Servizio, scegli quale servizio desideri utilizzare per archiviare il token e procedi come segue:
 - a. Se scegli di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova, quindi scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
 - b. Se scegli di utilizzarlo CodeBuild, inserisci il token di accesso personale di Bitbucket, quindi scegli Salva.
 - Seleziona Usa le credenziali di sostituzione solo per questo progetto per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni delle credenziali del tuo account.
 - a. Dall'elenco di credenziali compilato, scegli una delle opzioni in Token di accesso personale.
 - b. Puoi anche creare un nuovo token di accesso personale selezionando Crea una nuova connessione con token di accesso personale nella descrizione.

Connect Bitbucket con una password dell'app o un token di accesso (CLI)

Segui questi passaggi per utilizzare per connettere il tuo progetto AWS CLI a Bitbucket utilizzando una password o un token di accesso per l'app. Per informazioni sull'utilizzo di AWS CLI with AWS CodeBuild, consulta. [Guida di riferimento alla riga di comando](#)

1. Eseguire il comando `import-source-credentials`:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Nell'output vengono visualizzati dati in formato JSON. Copiare i dati in un file (ad esempio, *import-source-credentials.json*) in una posizione del computer o dell'istanza locale in cui AWS CLI è installato. Modificare i dati copiati come segue, quindi salvare i risultati.

```
{
  "serverType": "BITBUCKET",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Sostituisci quanto segue:

- *server-type*: Valore richiesto. Il provider di origine utilizzato per queste credenziali. I valori validi sono GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB e GITLAB_SELF_MANAGED.
- *auth-type*: Valore richiesto. Il tipo di autenticazione utilizzato per connettersi a un repository. I valori validi sono OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS e SECRETS_MANAGER. Perché, è consentito solo GitHub PERSONAL_ACCESS_TOKEN. BASIC_AUTH è consentito solo con la password dell'app Bitbucket.
- *should-overwrite*: Valore opzionale. Imposta su `false` per evitare di sovrascrivere le credenziali di origine del repository. Imposta su `true` per sovrascrivere le credenziali di origine del repository. Il valore predefinito è `true`.
- *token*: valore obbligatorio. Per il GitHub nostro GitHub Enterprise Server, questo è il token di accesso personale. Per Bitbucket, si tratta del token di accesso personale o della

password dell'app. Per il tipo di autenticazione CODECONNECTIONS, questo è l'ARN della connessione. Per il tipo di autenticazione SECRETS_MANAGER, questo è l'ARN segreto.

- **username**: Valore opzionale. Questo parametro viene ignorato per i provider GitHub di origine di GitHub Enterprise Server.
2. Per connettere il tuo account con una password dell'app o un token di accesso, passa alla directory che contiene il `import-source-credentials.json` file salvato nel passaggio 1 ed esegui nuovamente il `import-source-credentials` comando.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

I dati in formato JSON sono presenti nell'output con un Amazon Resource Name (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Se si esegue il comando `import-source-credentials` con lo stesso `server-type` e `auth-type` una seconda volta, il token di accesso archiviato viene aggiornato.

Dopo aver collegato l'account con una password per l'app, puoi utilizzarla `create-project` per creare il tuo CodeBuild progetto. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).

3. Per visualizzare le password o i token di accesso delle app connesse, esegui il `list-source-credentials` comando.

```
aws codebuild list-source-credentials
```

Un oggetto `sourceCredentialsInfos` in formato JSON è presente nell'output:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
```

```

        "serverType": "BITBUCKET",
        "arn": "arn"
    }
]
}

```

L'elemento `sourceCredentialsObject` contiene un elenco di informazioni sulle credenziali di origine connesse:

- L'elemento `authType` è il tipo di autenticazione utilizzato dalle credenziali. Può essere `OAUTH`, `BASIC_AUTH`, `PERSONAL_ACCESS_TOKEN`, `CODE_CONNECTIONS`, o `SECRETS_MANAGER`.
 - L'elemento `serverType` è il tipo di provider di origine. Questo può essere `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB`, o `GITLAB_SELF_MANAGED`.
 - L'elemento `arn` è l'ARN del token.
4. Per disconnetterti da un provider di origine e rimuovere la password dell'app o i token di accesso, esegui il `delete-source-credentials` comando con il relativo ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Vengono restituiti dati in formato JSON con l'ARN delle credenziali eliminate.

```

{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}

```

App Bitbucket OAuth

Connect Bitbucket tramite OAuth (console)

Per utilizzare la console per connettere il tuo progetto a Bitbucket tramite un' OAuth app, esegui le seguenti operazioni quando crei un progetto. Per informazioni, consulta [Creare un progetto di compilazione \(console\)](#).

1. In Source provider (Provider origine) selezionare Bitbucket.
2. Per Credential, esegui una delle seguenti operazioni:

- Scegli di utilizzare le credenziali dell'account per applicare le credenziali di origine predefinite dell'account a tutti i progetti.
 - a. Se non sei connesso a Bitbucket, scegli Gestisci le credenziali dell'account.
 - b. Per il tipo di credenziale, scegli l'app. OAuth
- Se hai scelto di utilizzare le credenziali a livello di account per il Servizio, scegli quale servizio desideri utilizzare per archiviare il token e procedi come segue:
 - a. Se scegli di utilizzare Secrets Manager, puoi scegliere di utilizzare una connessione segreta esistente o crearne una nuova, quindi scegliere Salva. Per ulteriori informazioni su come creare un nuovo segreto, consulta [Crea e archivia un token in un segreto di Secrets Manager](#).
 - b. Se scegli di utilizzare, CodeBuild quindi scegli Salva.
- Seleziona Usa le credenziali di sostituzione solo per questo progetto per utilizzare una credenziale di origine personalizzata per sovrascrivere le impostazioni delle credenziali del tuo account.
 - a. Dall'elenco di credenziali compilato, scegli una delle opzioni in app. OAuth
 - b. Puoi anche creare un nuovo token per OAuth l'app selezionando Crea una nuova connessione con il token dell'app OAuth nella descrizione.

Per esaminare le OAuth app autorizzate, accedi a [Autorizzazioni dell'applicazione](#) su Bitbucket e verifica che nell'elenco sia presente un'applicazione denominata. AWS CodeBuild (*region*)

GitLab accesso in CodeBuild

Infatti GitLab, si utilizza una GitLab connessione per accedere al provider di origine.

Argomenti

- [Connect CodeBuild a GitLab](#)

Connect CodeBuild a GitLab

Le connessioni consentono di autorizzare e stabilire configurazioni che associano il provider di terze parti all'utilizzo AWS delle risorse da parte dell'utente. AWS CodeConnections Per associare il tuo repository di terze parti come fonte per il tuo progetto di build, usi una connessione.

Per aggiungere un provider GitLab di sorgenti gestito GitLab CodeBuild autonomamente, puoi scegliere tra:

- Utilizza la CodeBuild console, la procedura guidata di creazione e compilazione del progetto o la pagina Modifica sorgente per scegliere l'opzione GitLab o il provider GitLab Self Managed. Vedi [Crea una connessione a GitLab \(console\)](#) per aggiungere il provider di origine. La console ti aiuta a creare una risorsa di connessioni.
- Usa la CLI per creare le tue risorse di connessione, vedi [Creare una connessione a GitLab \(CLI\)](#) per creare una risorsa di connessioni con la CLI.

 Note

Puoi anche creare una connessione utilizzando la console Developer Tools in Impostazioni. Vedi [Creare una connessione](#).

 Note

Autorizzando l'installazione di questa connessione in GitLab, concedi al nostro servizio le autorizzazioni per elaborare i tuoi dati accedendo al tuo account e puoi revocare le autorizzazioni in qualsiasi momento disinstallando l'applicazione.

Crea una connessione a GitLab

Questa sezione descrive come connettersi GitLab a CodeBuild. Per ulteriori informazioni sulle connessioni GitLab , consulta [Connect CodeBuild a GitLab](#).

Prima di iniziare:

- Devi aver già creato un account con GitLab.

 Note

Le connessioni forniscono l'accesso solo ai repository di proprietà dell'account utilizzato per creare e autorizzare la connessione.

 Note

È possibile creare connessioni a un repository in cui si ricopre il ruolo di proprietario e quindi la connessione può essere utilizzata con il repository con risorse come. GitLab CodeBuild Per i repository nei gruppi, non è necessario essere il proprietario del gruppo.

- Per specificare una fonte per il progetto di compilazione, devi aver già creato un repository su. GitLab

Argomenti

- [Crea una connessione a GitLab \(console\)](#)
- [Creare una connessione a GitLab \(CLI\)](#)

Crea una connessione a GitLab (console)

Segui questi passaggi per utilizzare la CodeBuild console e aggiungere una connessione per il tuo progetto (repository) in GitLab.

 Note

Invece di creare o utilizzare una connessione esistente nel tuo account, puoi utilizzare una connessione condivisa da un altro AWS account. Per ulteriori informazioni, consulta [Condividere le connessioni con AWS gli account](#).

Per creare o modificare il tuo progetto di compilazione

1. Accedi alla CodeBuild console.
2. Scegliere una delle seguenti opzioni.
 - Scegli di creare un progetto di compilazione. Segui i passaggi [Creare un progetto di compilazione \(console\)](#) per completare la prima schermata e nella sezione Sorgente, in Source Provider, scegli GitLab.
 - Scegli di modificare un progetto di build esistente. Scegliete Modifica, quindi scegliete Fonte. Nella pagina Modifica sorgente, in Provider di origine, scegli GitLab.

3. Seleziona una delle seguenti opzioni:

- In Connessione, scegli Connessione predefinita. La connessione predefinita applica una GitLab connessione predefinita a tutti i progetti.
- In Connessione, scegli Connessione personalizzata. La connessione personalizzata applica una GitLab connessione personalizzata che sostituisce le impostazioni predefinite dell'account.

4. Esegui una delle seguenti operazioni:

- In Connessione predefinita o Connessione personalizzata, se non hai già creato una connessione al tuo provider, scegli Crea una nuova GitLab connessione. Procedi al passaggio 5 per creare la connessione.
- In Connessione, se hai già creato una connessione al tuo provider, scegli la connessione. Procedi al passaggio 10.

 Note

Se si chiude la finestra pop-up prima di creare una GitLab connessione, è necessario aggiornare la pagina.

5. Per creare una connessione a un GitLab repository, in Seleziona un provider, scegli. GitLab In Connection name (Nome connessione), immetti il nome della connessione che desideri creare. Scegliere Connect to GitLab (Connetti ad Amazon Aurora).

[Developer Tools](#) > [Connections](#) > Create connection

Create a connection [Info](#)

Create GitLab connection [Info](#)

Connection name

► **Tags - optional**

Connect to GitLab

6. Quando viene GitLab visualizzata la pagina di accesso, accedi con le tue credenziali, quindi scegli Accedi.
7. Se è la prima volta che autorizzi la connessione, viene visualizzata una pagina di autorizzazione con un messaggio che richiede l'autorizzazione per la connessione per accedere al tuo account. GitLab

Seleziona Authorize (Autorizza).

Authorize **AWS Connector for GitLab** to use your account?

An application called **AWS Connector for GitLab** is requesting access to your GitLab account. This application was created by **Amazon AWS**. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**

Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.

- **Read the authenticated user's personal information**

Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

- **Read Api**

Grants read access to the API, including all groups and projects, the container registry, and the package registry.

- **Allows read-only access to the repository**

Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.

- **Allows read-write access to the repository**

Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

8. Il browser torna alla pagina della console delle connessioni. Nelle impostazioni di GitLab connessione, la nuova connessione viene mostrata in Nome connessione.
9. Scegli Connetti.

Dopo che una GitLab connessione è stata creata con successo, nella parte superiore verrà visualizzato un banner di successo.

10. Nella pagina Crea progetto di compilazione, nell'elenco a discesa Connessione predefinita o Connessione personalizzata, assicurati che l'ARN della connessione sia elencato. In caso contrario, scegli il pulsante di aggiornamento per farlo apparire.
11. In Repository, scegli il nome del tuo progetto in GitLab specificando il percorso del progetto con lo spazio dei nomi. Ad esempio, per un repository a livello di gruppo, inserisci il nome del repository nel seguente formato: `group-name/repository-name` [Per ulteriori informazioni sul percorso e sullo spazio dei nomi, consultate il campo in api/projects.html#path_with_namespace](https://docs.gitlab.com/ee/api/projects.html#path_with_namespace) <https://docs.gitlab.com/ee/get-single-project> [Per ulteriori informazioni sullo spazio dei nomi in, vedi user/namespace/](https://docs.gitlab.com/ee/user/namespace/). [GitLab https://docs.gitlab.com/ee/](https://docs.gitlab.com/ee/)

Note

Per i gruppi in GitLab, è necessario specificare manualmente il percorso del progetto con lo spazio dei nomi. Ad esempio, per un repository denominato `myrepo` in un gruppo `mygroup`, inserisci quanto segue: `mygroup/myrepo` Puoi trovare il percorso del progetto con lo spazio dei nomi nell'URL in. GitLab

12. Nella versione sorgente, facoltativa, inserisci un ID di pull request, branch, commit ID, tag o riferimento e un ID di commit. Per ulteriori informazioni, consulta [Esempio di versione sorgente con AWS CodeBuild](#).

Note

Ti consigliamo di scegliere nomi di branch Git che non assomiglino a commit IDs, come `811dd1ba1aba14473856cee38308caed7190c0d o5392f7`. Questo ti aiuta a evitare le collisioni tra i checkout di Git e i commit effettivi.

13. In Git clone depth, opzionale, puoi creare un clone poco profondo con una cronologia troncata al numero di commit specificato. Se si desidera un clone completo, selezionare Full (Completo).
14. In Build Status - opzionale, seleziona Segnala lo stato della build al fornitore di origine all'inizio e alla fine delle build se desideri che lo stato dell'inizio e del completamento della build venga segnalato al tuo fornitore di origine.

Per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve avere accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Creare una connessione a GitLab (CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per creare una connessione.

Per farlo, utilizzare il comando create-connection.

Important

Per impostazione predefinita, una connessione creata tramite AWS CLI o AWS CloudFormation è in PENDING stato. Dopo aver creato una connessione con la CLI o CloudFormation, utilizza la console per modificare la connessione e definirne lo stato. AVAILABLE

Come creare una connessione

- Segui le istruzioni nella Guida per l'utente della console Developer Tools per [creare una connessione a GitLab \(CLI\)](#).

Prevenzione del confused deputy tra servizi

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel frattempo AWS, l'impersonificazione tra servizi può portare alla confusione del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente,

a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse dell'account.

Si consiglia di utilizzare `aws:SourceArn` le chiavi di contesto della condizione `aws:SourceAccount` globale nelle politiche delle risorse per limitare le autorizzazioni che AWS CodeBuild forniscono un altro servizio alla risorsa. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il modo più efficace per proteggersi dal problema “confused deputy” è quello di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:codebuild:*:123456789012:*`.

Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni.

Il valore di `aws:SourceArn` deve essere l' CodeBuild ARN del progetto.

L'esempio seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition CodeBuild per evitare il confuso problema del vice.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:us-east-1:111122223333:project/MyProject"
        }
      }
    }
  ]
}
```

```
}  
  ]  
    }  
      }  
        }
```

Argomenti avanzati

Questa sezione include diversi argomenti avanzati che sono utili per utente AWS CodeBuild più esperti.

Argomenti

- [Consenti agli utenti di interagire con CodeBuild](#)
- [Consenti CodeBuild di interagire con altri servizi AWS](#)
- [Crittografa gli output della build utilizzando una chiave gestita dal cliente](#)
- [Interagisci con l' CodeBuild utilizzo di AWS CLI](#)
- [Riferimento alla riga di comando per AWS CodeBuild](#)
- [AWS SDKs e strumenti di riferimento per AWS CodeBuild](#)
- [Utilizzo di questo servizio con un AWS SDK](#)
- [Specificare l' AWS CodeBuild endpoint](#)
- [Usa AWS CodeBuild with AWS CodePipeline per testare il codice ed eseguire build](#)
- [Uso AWS CodeBuild con Codecov](#)
- [Uso AWS CodeBuild con Jenkins](#)
- [Utilizzo AWS CodeBuild con applicazioni serverless](#)
- [Avvisi di terze parti AWS CodeBuild per Windows](#)
- [Usa le chiavi di CodeBuild condizione come variabili del ruolo del servizio IAM per controllare l'accesso alla build](#)
- [AWS CodeBuild tasti di condizione](#)

Consenti agli utenti di interagire con CodeBuild

Se segui la procedura di accesso [Nozioni di base utilizzando la console](#) AWS CodeBuild per la prima volta, probabilmente non hai bisogno delle informazioni contenute in questo argomento. Tuttavia, continuando a utilizzare CodeBuild, potresti voler fare cose come dare ad altri utenti e gruppi dell'organizzazione la possibilità di interagire con CodeBuild.

Per consentire a un utente o un gruppo IAM di interagire con AWS CodeBuild, devi concedere loro le autorizzazioni di accesso a CodeBuild. Questa sezione descrive come eseguire questa operazione con la console IAM o il AWS CLI.

Se accederai CodeBuild con il tuo account AWS root (scelta non consigliata) o con un utente amministratore nel tuo AWS account, non è necessario seguire queste istruzioni.

Per informazioni sugli account AWS root e sugli utenti amministratori, vedere [L'utente Account AWS root](#) e [Creare il primo utente e gruppo Account AWS root](#) nella Guida per l'utente.

Per aggiungere le autorizzazioni di CodeBuild accesso a un gruppo o utente IAM (console)

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

Dovresti aver già effettuato l'accesso Console di gestione AWS utilizzando uno dei seguenti:

- Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, consulta [L'utente Account AWS root](#) nella guida per l'utente.
- Un utente amministratore nel tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a eseguire il seguente set minimo di azioni:

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam:ListAttachedGroupPolicies
iam:ListAttachedUserPolicies
iam:ListGroups
iam:ListPolicies
iam:ListUsers
```

Per ulteriori informazioni, consulta [Panoramica delle politiche IAM](#) nella Guida per l'utente.

2. Nel riquadro di navigazione, scegli Policy.
3. Per aggiungere un set personalizzato di autorizzazioni di AWS CodeBuild accesso a un gruppo IAM o a un utente IAM, vai avanti al passaggio 4 di questa procedura.

Per aggiungere un set predefinito di autorizzazioni di CodeBuild accesso a un gruppo IAM o a un utente IAM, scegli Policy Type, AWS Managed, quindi procedi come segue:

- Per aggiungere le autorizzazioni di accesso complete a CodeBuild, seleziona la casella denominata AWSCodeBuildAdminAccess, scegli Policy Actions, quindi scegli Allega. Seleziona la casella accanto al gruppo o all'utente IAM di destinazione, quindi scegli Allega

policy. Ripeti questa operazione per le politiche denominate AmazonS3 ReadOnlyAccess e Access. IAMFull

- Per aggiungere le autorizzazioni di accesso a tutto CodeBuild tranne l'amministrazione del progetto di compilazione, seleziona la casella denominata AWSCodeBuildDeveloperAccess, scegli Policy Actions, quindi scegli Allega. Seleziona la casella accanto al gruppo o all'utente IAM di destinazione, quindi scegli Allega policy. Ripeti questa operazione per la politica denominata ReadOnlyAccessAmazonS3.
- Per aggiungere autorizzazioni di accesso in sola lettura, seleziona le caselle denominate CodeBuild. AWSCodeBuildReadOnlyAccess Seleziona la casella accanto al gruppo o all'utente IAM di destinazione, quindi scegli Allega policy. Ripeti questa operazione per la politica denominata ReadOnlyAccessAmazonS3.

Ora hai aggiunto un set predefinito di autorizzazioni di CodeBuild accesso a un gruppo o utente IAM. Salta il resto dei passaggi in questa procedura.

4. Scegli Crea policy.
5. Nella pagina Create Policy (Crea policy), accanto a Create Your Own Policy (Crea la tua policy), selezionare Select (Seleziona).
6. Nella pagina Review Policy (Rivedi policy), immettere un nuovo nome per la policy in Policy Name (Nome policy), ad esempio **CodeBuildAccessPolicy**. Se si utilizza un nome diverso, assicurarsi di ripeterlo in tutta questa procedura.
7. Per Policy Document (Documento policy) immettere quanto indicato di seguito e quindi scegliere Create Policy (Crea policy).

JSON

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



```
        "Sid": "CodeBuildRolePolicy",
        "Effect": "Allow",
        "Action": [
            "iam:PassRole"
        ],
        "Resource": "arn:aws:iam::111122223333:role/role-name"
    },
    {
        "Sid": "CloudWatchLogsAccessPolicy",
        "Effect": "Allow",
        "Action": [
            "logs:FilterLogEvents",
            "logs:GetLogEvents"
        ],
        "Resource": "*"
    },
    {
        "Sid": "S3AccessPolicy",
        "Effect": "Allow",
        "Action": [
            "s3:CreateBucket",
            "s3:GetObject",
            "s3:List*",
            "s3:PutObject"
        ],
        "Resource": "*"
    },
    {
        "Sid": "S3BucketIdentity",
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketAcl",
            "s3:GetBucketLocation"
        ],
        "Resource": "*"
    }
]
}
```

 Note

Questa policy consente l'accesso a tutte CodeBuild le azioni e a un numero potenzialmente elevato di AWS risorse. Per limitare le autorizzazioni a CodeBuild azioni specifiche, modifica il valore di `codebuild:*` nella dichiarazione CodeBuild politica. Per ulteriori informazioni, consulta [Gestione dell'identità e degli accessi](#). Per limitare l'accesso a AWS risorse specifiche, modificate il valore dell'`Resource` oggetto. Per ulteriori informazioni, consulta [Gestione dell'identità e degli accessi](#).

8. Nel riquadro di navigazione selezionare Groups (Gruppi) o Users (Utenti).
9. Nell'elenco dei gruppi o degli utenti, scegli il nome del gruppo IAM o dell'utente IAM a cui desideri aggiungere le autorizzazioni di CodeBuild accesso.
10. Per un gruppo, nella pagina relativa alle impostazioni del gruppo, nella scheda Permissions (Autorizzazioni), espandere la sezione Managed Policies (Policy gestite) e selezionare Attach Policy (Collega policy).

Per un utente, nella pagina di impostazioni utente, nella scheda Permissions (Autorizzazioni), selezionare Add permissions (Aggiungi autorizzazioni).

11. Per un gruppo, nella pagina Allega policy, seleziona CodeBuildAccessPolicy, quindi scegli Allega policy.

Per un utente, nella pagina Aggiungi autorizzazioni, scegli Allega direttamente le politiche esistenti. Seleziona CodeBuildAccessPolicy, scegli Avanti: Revisione, quindi scegli Aggiungi autorizzazioni.

Per aggiungere le autorizzazioni di CodeBuild accesso a un gruppo o utente IAM ()AWS CLI

1. Assicurati di averlo configurato AWS CLI con la chiave di AWS accesso e la chiave di accesso AWS segreta che corrispondono a una delle entità IAM, come descritto nella procedura precedente. Per ulteriori informazioni, consulta [Come configurare AWS Command Line Interface](#) nella Guida per l'utente di AWS Command Line Interface .
2. Per aggiungere un set personalizzato di autorizzazioni di AWS CodeBuild accesso a un gruppo IAM o a un utente IAM, vai al passaggio 3 di questa procedura.

Per aggiungere un set predefinito di autorizzazioni di CodeBuild accesso a un gruppo IAM o a un utente IAM, procedi come segue:

Esegui uno dei seguenti comandi, a seconda che desideri aggiungere autorizzazioni a un gruppo o utente IAM:

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn  
  
aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

È necessario eseguire il comando tre volte, sostituendo *group-name* o *user-name* con il nome del gruppo o il nome utente IAM e sostituendo *policy-arn* una volta per ciascuna delle seguenti policy Amazon Resource Names (ARNs):

- Per aggiungere le autorizzazioni di accesso complete a CodeBuild, utilizza la seguente politica ARNs:
 - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
 - `arn:aws:iam::aws:policy/IAMFullAccess`
- Per aggiungere le autorizzazioni di accesso a tutto CodeBuild tranne l'amministrazione del progetto di compilazione, utilizza la seguente politica: ARNs
 - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Per aggiungere autorizzazioni di accesso di sola lettura a CodeBuild, utilizza la seguente politica: ARNs
 - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Ora hai aggiunto un set predefinito di autorizzazioni di CodeBuild accesso a un gruppo o utente IAM. Salta il resto dei passaggi in questa procedura.

3. In una directory vuota della workstation o dell'istanza locale in cui AWS CLI è installato, crea un file denominato `put-group-policy.json` o `put-user-policy.json`. Se si utilizza un nome file diverso, assicurarsi di ripeterlo in tutta questa procedura.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "codebuild:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeBuildRolePolicy",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::111122223333:role/role-name"
    },
    {
      "Sid": "CloudWatchLogsAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3AccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:GetObject",
        "s3:List*",
        "s3:PutObject"
      ],
      "Resource": "*"
    }
  ]
}
```

```
        "Sid": "S3BucketIdentity",
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketAcl",
            "s3:GetBucketLocation"
        ],
        "Resource": "*"
    }
}
```

Note

Questa politica consente l'accesso a tutte CodeBuild le azioni e a un numero potenzialmente elevato di AWS risorse. Per limitare le autorizzazioni a CodeBuild azioni specifiche, modifica il valore di `codebuild:*` nella dichiarazione CodeBuild politica. Per ulteriori informazioni, consulta [Gestione dell'identità e degli accessi](#). Per limitare l'accesso a AWS risorse specifiche, modificate il valore dell'`Resource` oggetto correlato. Per ulteriori informazioni, consulta [Gestione dell'identità e degli accessi](#) o la documentazione sulla sicurezza del servizio AWS specifico.

4. Passare alla directory contenente il file salvato, quindi eseguire uno dei seguenti comandi. Puoi utilizzare valori diversi per `CodeBuildGroupAccessPolicy` e `CodeBuildUserAccessPolicy`. Se si utilizzano valori diversi, assicurarsi di specificarli qui.

Per un gruppo IAM:

```
aws iam put-group-policy --group-name group-name --policy-name
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Per un utente :

```
aws iam put-user-policy --user-name user-name --policy-name
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

Nei comandi precedenti, sostituisci *group-name* o *user-name* con il nome del gruppo o dell'utente IAM di destinazione.

Consenti CodeBuild di interagire con altri servizi AWS

Se segui la procedura di accesso [Nozioni di base utilizzando la console](#) AWS CodeBuild per la prima volta, probabilmente non hai bisogno delle informazioni contenute in questo argomento. Tuttavia, continuando a utilizzare CodeBuild, potresti voler fare cose come consentire CodeBuild l'interazione con altri AWS servizi.

CodeBuild Per consentire l'interazione con i AWS servizi dipendenti per conto dell'utente, è necessario un ruolo AWS CodeBuild di servizio. È possibile creare un ruolo di CodeBuild servizio utilizzando le AWS CodePipeline console CodeBuild or. Per informazioni, consulta:

- [Creare un progetto di compilazione \(console\)](#)
- [Crea una pipeline che utilizza CodeBuild \(CodePipeline console\)](#)
- [Aggiungere un'azione di CodeBuild compilazione a una pipeline \(CodePipeline console\)](#)
- [Modifica delle impostazioni di un progetto di compilazione \(console\)](#)

Se non prevedi di utilizzare queste console, questa sezione descrive come creare un ruolo di CodeBuild servizio con la console IAM o il AWS CLI

Important

CodeBuild utilizza il ruolo di servizio per tutte le operazioni eseguite per tuo conto. Se il ruolo include le autorizzazioni di cui l'utente non dovrebbe disporre, puoi riassegnare involontariamente le autorizzazioni di un utente. Assicurati che il ruolo garantisca i [privilegi minimi](#).

Il ruolo del servizio descritto in questa pagina contiene una policy che concede le autorizzazioni minime richieste per utilizzare CodeBuild. Potrebbe essere necessario aggiungere autorizzazioni aggiuntive, a seconda del caso d'uso.

Per creare un ruolo CodeBuild di servizio (console)

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

È necessario aver effettuato l'accesso alla console utilizzando uno dei seguenti metodi:

- Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, consulta [L'utente Account AWS root](#) nella guida per l'utente.

- Un utente amministratore nel tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a eseguire il seguente set minimo di azioni:

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam:ListAttachedRolePolicies
iam:ListPolicies
iam:ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Per ulteriori informazioni, consulta [Panoramica delle politiche IAM](#) nella Guida per l'utente.

2. Nel riquadro di navigazione, scegli Policy.
3. Scegli Crea policy.
4. Nella pagina Create policy (Crea policy), selezionare JSON.
5. Per la policy JSON, immettere quanto segue, quindi selezionare Review Policy (Rivedi policy):

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ],
}
```

```
{
  "Sid": "CodeCommitPolicy",
  "Effect": "Allow",
  "Action": [
    "codecommit:GitPull"
  ],
  "Resource": "*"
},
{
  "Sid": "S3GetObjectPolicy",
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion"
  ],
  "Resource": "*"
},
{
  "Sid": "S3PutObjectPolicy",
  "Effect": "Allow",
  "Action": [
    "s3:PutObject"
  ],
  "Resource": "*"
},
{
  "Sid": "ECRPullPolicy",
  "Effect": "Allow",
  "Action": [
    "ecr:BatchCheckLayerAvailability",
    "ecr:GetDownloadUrlForLayer",
    "ecr:BatchGetImage"
  ],
  "Resource": "*"
},
{
  "Sid": "ECRAuthPolicy",
  "Effect": "Allow",
  "Action": [
    "ecr:GetAuthorizationToken"
  ],
  "Resource": "*"
},
{
```



```
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

 Note

Questa policy contiene istruzioni che consentono l'accesso a un numero potenzialmente elevato di AWS risorse. Per limitare l'accesso AWS CodeBuild a AWS risorse specifiche, modificate il valore dell'`Resource` array. Per ulteriori informazioni, consulta la documentazione sulla sicurezza del AWS servizio.

6. Nella pagina Review policy (Rivedi policy), in Policy Name (Nome policy), immettere un nome per la policy (ad esempio **CodeBuildServiceRolePolicy**), quindi selezionare Create policy (Crea policy).

 Note

Se si utilizza un nome diverso, assicurarsi di ripeterlo in tutta questa procedura.

7. Nel pannello di navigazione, seleziona Roles (Ruoli).
8. Selezionare Create role (Crea ruolo).
9. Nella pagina Crea ruolo, con AWS Servizio già selezionato, scegli CodeBuild, quindi scegli Avanti: Autorizzazioni.
10. Nella pagina Allega criteri di autorizzazione, seleziona CodeBuildServiceRolePolicy, quindi scegli Avanti: revisione.
11. Nella pagina Create role and review (Crea ruolo e rivedi), in Role name (Nome ruolo), immettere un nome per il ruolo (ad esempio, **CodeBuildServiceRole**), quindi selezionare Create role (Crea ruolo).

Per creare un ruolo CodeBuild di servizio (AWS CLI)

1. Assicurati di averlo configurato AWS CLI con la chiave di AWS accesso e la chiave di accesso AWS segreta che corrispondono a una delle entità IAM, come descritto nella procedura precedente. Per ulteriori informazioni, consulta [Come configurare AWS Command Line Interface](#) nella Guida per l'utente di AWS Command Line Interface .
2. In una directory vuota della workstation o dell'istanza locale in cui AWS CLI è installato, crea due file denominati `create-role.json` and `put-role-policy.json`. Se si utilizzano nomi file diversi, assicurarsi di ripeterli in tutta questa procedura.

`create-role.json`:

JSON

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

Note

Si consiglia di utilizzare le chiavi di condizione `aws:SourceAccount` e `aws:SourceArn` per proteggersi dal [problema del "confused deputy"](#). Ad esempio, è possibile modificare la precedente politica di attendibilità con i seguenti blocchi di condizioni. Il `aws:SourceAccount` è il proprietario del CodeBuild progetto e il CodeBuild progetto `aws:SourceArn` è l'ARN.

Se desideri limitare il tuo ruolo di servizio a un AWS account, `create-role.json` potresti avere un aspetto simile a questo:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "111122223333"
          ]
        }
      }
    }
  ]
}
```

Se desideri limitare il tuo ruolo di servizio a un CodeBuild progetto specifico, `create-role.json` potresti avere un aspetto simile a questo:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
```

```

        "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject"
    }
}
]
}

```

Note

Se non conosci o non hai ancora deciso un nome per il tuo CodeBuild progetto e desideri una restrizione della politica di trust su un particolare pattern ARN, puoi sostituire quella parte dell'ARN con un carattere jolly (*). Dopo aver creato il progetto, puoi aggiornare la politica di attendibilità.

put-role-policy.json:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    }
  ]
}

```

```
    },
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3PutObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3BucketIdentity",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Questa politica contiene istruzioni che consentono l'accesso a un numero potenzialmente elevato di AWS risorse. Per limitare l'accesso AWS CodeBuild a AWS risorse specifiche, modificate il valore dell'`Resourcearray`. Per ulteriori informazioni, consulta la documentazione sulla sicurezza del AWS servizio.

3. Passare alla directory in cui sono stati salvati i file precedenti, quindi eseguire i seguenti due comandi, uno alla volta, in questo ordine. È possibile utilizzare valori diversi per `CodeBuildServiceRole` e `CodeBuildServiceRolePolicy`, ma assicurarsi di specificarli qui.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document  
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name  
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

Crittografa gli output della build utilizzando una chiave gestita dal cliente

Se segui i passaggi AWS CodeBuild per accedere [Nozioni di base utilizzando la console](#) per la prima volta, molto probabilmente non hai bisogno delle informazioni contenute in questo argomento. Tuttavia, continuando a utilizzare CodeBuild, potresti voler eseguire operazioni come crittografare gli elementi della build.

Per AWS CodeBuild crittografare gli artefatti di output della build, deve accedere a una chiave KMS. Per impostazione predefinita, CodeBuild utilizza il Chiave gestita da AWS per Amazon S3 nel tuo AWS account.

Se non desideri utilizzare il Chiave gestita da AWS, devi creare e configurare tu stesso una chiave gestita dal cliente. Questa sezione descrive come eseguire questa operazione con la console IAM.

Per informazioni sulle chiavi gestite dai clienti, consulta [AWS Key Management Service Concepts](#) and [Creating Keys](#) nella AWS KMS Developer Guide.

Per configurare una chiave gestita dal cliente da utilizzare da parte di CodeBuild, segui le istruzioni nella sezione «Come modificare una politica chiave» della sezione [Modificare una politica chiave](#) nella Guida per gli AWS KMS sviluppatori. Quindi aggiungi le seguenti istruzioni (tra **### BEGIN ADDING STATEMENTS HERE ###** e **### END ADDING STATEMENTS HERE ###**) alla politica chiave. Le ellissi (. . .) vengono utilizzate per brevità e per aiutare i clienti a individuare il punto in cui aggiungere le istruzioni. Non rimuovere nessuna istruzione e non digitare queste ellissi nella policy della chiave.

JSON

```
{  
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.us-east-1.amazonaws.com",
        "kms:CallerAccount": "111122223333"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
]
```

- **region-ID** rappresenta l'ID della AWS regione in cui si CodeBuild trovano i bucket Amazon S3 associati (ad esempio,). us-east-1
- **account-ID** rappresenta l'ID dell' AWS account che possiede la chiave gestita dal cliente.
- **CodeBuild-service-role** rappresenta il nome del ruolo di CodeBuild servizio creato o identificato in precedenza in questo argomento.

 Note

Per creare o configurare una chiave gestita dal cliente tramite la console IAM, devi prima accedere a Console di gestione AWS utilizzando uno dei seguenti:

- Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, consulta [The Account Root User](#) nella guida per l'utente.
- Un utente amministratore del tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a creare o modificare la chiave gestita dal cliente. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per utilizzare la AWS KMS console](#) nella Guida per gli AWS KMS sviluppatori.

Interagisci con l' CodeBuild utilizzo di AWS CLI

Se segui la procedura di accesso [Nozioni di base utilizzando la console](#) AWS CodeBuild per la prima volta, probabilmente non hai bisogno delle informazioni contenute in questo argomento. Tuttavia, continuando a utilizzare CodeBuild, potresti voler eseguire operazioni come consentire agli utenti di utilizzare il AWS CLI per CodeBuild interagire con (o in aggiunta) alla CodeBuild console, alla CodePipeline console o al AWS SDKs.

Per installare e configurare AWS CLI, consulta [Getting Set Up with the AWS Command Line Interface](#) nella Guida AWS Command Line Interface per l'utente.

Dopo aver installato AWS CLI, completa le seguenti attività:

1. Esegui il comando seguente per confermare se l'installazione dei AWS CLI supporti CodeBuild:

```
aws codebuild list-builds
```

Se eseguito correttamente, delle informazioni simili alle seguenti appariranno nell'output:

```
{
  "ids": []
}
```

Le parentesi quadre vuote indicano che non hai ancora eseguito compilazioni.

2. In caso di errore, devi disinstallare la versione attuale di AWS CLI e installare l'ultima versione. Per ulteriori informazioni, consulta [Disinstallare AWS CLI](#) e [Installare AWS Command Line Interface](#) nella Guida per l'utente di AWS Command Line Interface .

Riferimento alla riga di comando per AWS CodeBuild

AWS CLI Fornisce comandi per l'automazione AWS CodeBuild. Utilizza le informazioni contenute in questo argomento come supplemento alla [Guida per l'AWS Command Line Interface utente](#) e al [AWS CLI Reference for AWS CodeBuild](#).

Non hai trovato ciò che cerchi? Se desideri utilizzare AWS SDKs to call CodeBuild, consulta il [AWS SDKs e riferimento agli strumenti](#).

Per utilizzare le informazioni contenute in questo argomento, è necessario averle già installate AWS CLI e configurate per l'uso con CodeBuild, come descritto in [Interagisci con l' CodeBuild utilizzo di AWS CLI](#).

Per utilizzare il AWS CLI per specificare l'endpoint CodeBuild, vedere [Specificate l' AWS CodeBuild endpoint \(\)AWS CLI](#).

Eseguite questo comando per ottenere un elenco di CodeBuild comandi.

```
aws codebuild help
```

Esegui questo comando per ottenere informazioni su un CodeBuild comando, *command-name* dov'è il nome del comando.

```
aws codebuild command-name help
```

CodeBuild i comandi includono:

- `batch-delete-builds`: elimina una o più build. CodeBuild Per ulteriori informazioni, consulta [Eliminazione delle compilazioni \(AWS CLI\)](#).
- `batch-get-builds`: consente di ottenere informazioni su più compilazioni in CodeBuild. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli delle compilazioni \(AWS CLI\)](#).
- `batch-get-projects`: consente di ottenere informazioni su uno o più progetti di compilazione specificati. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un progetto di compilazione \(AWS CLI\)](#).

- `create-project`: consente di creare un progetto di compilazione. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione \(AWS CLI\)](#).
- `delete-project`: consente di eliminare un progetto di compilazione. Per ulteriori informazioni, consulta [Eliminazione di un progetto di compilazione \(AWS CLI\)](#).
- `list-builds`: elenca Amazon Resource Names (ARNs) per le build in CodeBuild. Per ulteriori informazioni, consulta [Visualizza un elenco di build \(\) IDs AWS CLI](#).
- `list-builds-for-project`: ottiene un elenco di build IDs associate a un progetto di build specificato. Per ulteriori informazioni, consulta [Visualizza un elenco di build IDs per un progetto di compilazione \(AWS CLI\)](#).
- `list-curated-environment-images`: ottiene un elenco di immagini Docker gestite da CodeBuild che puoi usare per le tue build. Per ulteriori informazioni, consulta [Immagini Docker fornite da CodeBuild](#).
- `list-projects`: consente di ottenere l'elenco dei nomi dei progetti di compilazione. Per ulteriori informazioni, consulta [Visualizzazione di un elenco di nomi di progetti di compilazione \(AWS CLI\)](#).
- `start-build`: consente di avviare l'esecuzione di una compilazione. Per ulteriori informazioni, consulta [Esecuzione di una compilazione \(AWS CLI\)](#).
- `stop-build`: consente di tentare l'interruzione dell'esecuzione della compilazione specificata. Per ulteriori informazioni, consulta [Interruzione di una compilazione \(AWS CLI\)](#).
- `update-project`: consente di modificare le informazioni sul progetto di compilazione specificato. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#).

AWS SDKs e strumenti di riferimento per AWS CodeBuild

Per utilizzare uno degli strumenti AWS SDKs di automazione AWS CodeBuild, consulta le seguenti risorse.

Se desideri utilizzare AWS CLI to run CodeBuild, consulta il [Guida di riferimento alla riga di comando](#).

Supportato AWS SDKs e strumenti per AWS CodeBuild

Quanto segue AWS SDKs e supporta gli strumenti CodeBuild:

- SDK [AWS per C++](#). Per ulteriori informazioni, consulta la sezione [Aws::CodeBuild](#) namespace del riferimento all'API AWS SDK for C++.

- SDK [AWS per Go](#). Per ulteriori informazioni, consulta la sezione [codebuild](#) dell'AWS SDK for Go API Reference.
- SDK [AWS per Java](#). Per ulteriori informazioni, consultare le sezioni `com.amazonaws.services.codebuild` e `com.amazonaws.services.codebuild.model` dell'[SDK AWS del riferimento API per Java](#).
- L'[AWS SDK per JavaScript il browser e l'AWSSDK](#) per Node.js. JavaScript Per ulteriori informazioni, consulta la [Classe: AWS CodeBuild](#) sezione dell'AWS SDK for JavaScript API Reference.
- SDK [AWS per .NET](#). Per ulteriori informazioni consultare le sezioni dello spazio dei nomi [Amazon.CodeBuild](#) e [Amazon.CodeBuild.Model](#) di AWS SDK per il riferimento API di .NET.
- SDK [AWS per PHP](#). Per ulteriori informazioni, consultare la sezione [Namespace Aws\CodeBuild](#) dell'AWS SDK per il riferimento API di PHP.
- SDK [AWS per Python \(Boto3\)](#). Per ulteriori informazioni, consultare la sezione [CodeBuild](#) della Documentazione Boto 3.
- SDK [AWS per Ruby](#). Per ulteriori informazioni, consultare la sezione [Module: AWS::CodeBuild](#) dell'SDK AWS per il riferimento API di Ruby.
- Gli [AWS strumenti per PowerShell](#). Per ulteriori informazioni, vedere la [AWS CodeBuild](#) sezione AWS Tools for PowerShell Cmdlet Reference.

Utilizzo di questo servizio con un AWS SDK

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK per Rust	AWS SDK per Rust esempi di codice
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Per esempi specifici del servizio, consulta [Esempi di codice per CodeBuild l'utilizzo AWS SDKs](#).

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link
Fornisci un feedback nella parte inferiore di questa pagina.

Specificare l' AWS CodeBuild endpoint

È possibile utilizzare il AWS Command Line Interface (AWS CLI) o uno dei AWS SDKs per specificare l'endpoint utilizzato da AWS CodeBuild. Esiste un endpoint per ogni regione in cui CodeBuild è disponibile. Oltre a un endpoint regionale, quattro regioni hanno un endpoint FIPS (Federal Information Processing Standards). Per ulteriori informazioni sugli endpoint FIPS, consulta la [panoramica su FIPS 140-2](#).

La specifica di un endpoint è facoltativa. Se non indichi esplicitamente CodeBuild quale endpoint utilizzare, il servizio utilizza l'endpoint associato alla regione utilizzata dal tuo account. AWS

CodeBuild per impostazione predefinita non viene mai utilizzato un endpoint FIPS. Se desideri utilizzare un endpoint FIPS, devi associarvi CodeBuild utilizzando uno dei seguenti metodi.

Note

È possibile utilizzare un alias o un nome di regione per specificare un endpoint utilizzando un SDK. AWS Se si utilizza il AWS CLI, è necessario utilizzare il nome completo dell'endpoint.

Per gli endpoint che possono essere utilizzati con CodeBuild, vedi [CodeBuild regioni ed endpoint](#).

Argomenti

- [Specificate l' AWS CodeBuild endpoint \(AWS CLI\)](#)
- [Specificare l' AWS CodeBuild endpoint \(AWS SDK\)](#)

Specificate l' AWS CodeBuild endpoint (AWS CLI)

È possibile utilizzare il AWS CLI per specificare l'endpoint tramite il quale AWS CodeBuild si accede utilizzando l'`--endpoint-url` argomento in qualsiasi CodeBuild comando. Ad esempio, esegui questo comando per ottenere un elenco di nomi di build di progetti utilizzando l'endpoint FIPS (Federal Information Processing Standards) nella regione Stati Uniti orientali (Virginia settentrionale):

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Includi `https://` all'inizio dell'endpoint.

L'`--endpoint-url` AWS CLI argomento è disponibile per tutti i servizi. AWS Per ulteriori informazioni su questo e altri AWS CLI argomenti, vedere [AWS CLI Command Reference](#).

Specificare l' AWS CodeBuild endpoint (AWS SDK)

È possibile utilizzare un AWS SDK per specificare l'endpoint tramite il quale si accede. AWS CodeBuild Sebbene questo esempio utilizzi l'[AWS SDK for Java](#), puoi specificare l'endpoint con l'altro. AWS SDKs

Utilizzate il `withEndpointConfiguration` metodo per costruire il client Build. AWSCode Il formato da utilizzare è il seguente:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
    "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Per informazioni su `AWSCodeBuildClientBuilder`, vedete [Class AWSCode BuildClientBuilder](#).

Le credenziali utilizzate in `withCredentials` devono essere di tipo `AWSCredentialsProvider`. Per ulteriori informazioni, consulta [Lavorare con AWS le credenziali](#).

Non includere `https://` all'inizio dell'endpoint.

Se desideri specificare un endpoint non FIPS, puoi usare la regione anziché l'endpoint effettivo. Ad esempio, per specificare l'endpoint nella regione Stati Uniti orientali (Virginia settentrionale), puoi utilizzare `us-east-1` al posto del nome completo dell'endpoint, `codebuild.us-east-1.amazonaws.com`.

Se desideri specificare un endpoint FIPS, puoi usare un alias per semplificare il codice. Solo gli endpoint FIPS hanno un alias. Altri endpoint devono essere specificati utilizzando la regione o il nome completo.

Nella tabella seguente sono elencati gli alias di ognuno dei quattro endpoint FIPS disponibili.

Nome Regione	Regione	Endpoint	Alias
US East (N. Virginia)	us-east-1	codebuild-fips.us-east-1.amazonaws.com	us-east-1-fips
Stati Uniti orientali (Ohio)	us-east-2	codebuild-fips.us-east-2.amazonaws.com	us-east-2-fips
US West (N. California)	us-west-1	codebuild-fips.us-west-1.amazonaws.com	us-west-1-fips

Nome Regione	Regione	Endpoint	Alias
US West (Oregon)	us-west-2	codebuild-fips.us-west-2.amazonaws.com	us-west-2-fips

Per specificare l'uso dell'endpoint FIPS nella regione Stati Uniti occidentali (Oregon) utilizzando un alias:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Per specificare l'uso dell'endpoint non FIPS nella regione Stati Uniti orientali (Virginia settentrionale):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
"us-east-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Per specificare l'uso dell'endpoint non FIPS nella regione Asia Pacifico (Mumbai):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
"ap-south-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Usa AWS CodeBuild with AWS CodePipeline per testare il codice ed eseguire build

Puoi automatizzare il processo di rilascio utilizzandolo AWS CodePipeline per testare il codice ed eseguire le build. AWS CodeBuild

La tabella seguente elenca le attività e i metodi disponibili per eseguirle. L'utilizzo AWS SDKs di per eseguire queste attività non rientra nell'ambito di questo argomento.

Operazione	Approcci disponibili	Approcci descritti in questo argomento
Crea una pipeline di distribuzione continua (CD) con CodePipeline cui automatizzare le build con CodeBuild	• CodePipeline console • AWS CLI • AWS SDKs	• Utilizzo della console CodePipeline • Utilizzo dell' AWS CLI • È possibile adattare le informazioni contenute in questo argomento per utilizzare il AWS SDKs. Per ulteriori informazioni, consulta la documentazione sulle <code>create-pipeline</code> azioni per il tuo linguaggio di programmazione nella SDKs sezione Tools for Amazon Web Services o consulta la CreatePipeline Guida di riferimento alle AWS CodePipeline API.
Aggiungi test e crea automazione con CodeBuild a una pipeline esistente in CodePipeline	• CodePipeline console • AWS CLI • AWS SDKs	• Utilizzo della console CodePipeline per aggiungere l'automazione delle compilazioni • Utilizzo della console CodePipeline per aggiungere l'automazione dei test • Per AWS CLI, è possibile adattare le informazioni contenute in questo argomento per creare una pipeline che contenga un'azione di CodeBuild compilazione o un'azione di test. Per ulteriori informazioni, vedete Edit a pipeline (AWS CLI) e il riferimento alla struttura della CodePipeline pipeline nella Guida per l'AWS CodePipeline utente. • È possibile adattare le informazioni contenute in questo argomento per utilizzare il. AWS SDKs Per ulteriori informazioni, consulta la documentazione sulle

Operazione	Approcci disponibili	Approcci descritti in questo argomento
		<code>update-pipeline</code> azioni per il tuo linguaggio di programmazione nella SDKs sezione Tools for Amazon Web Services o consulta la UpdatePipeline Guida di riferimento alle AWS CodePipeline API.

Argomenti

- [Prerequisiti](#)
- [Crea una pipeline che utilizza CodeBuild \(CodePipeline console\)](#)
- [Crea una pipeline che utilizza CodeBuild \(\)AWS CLI](#)
- [Aggiungere un'azione di CodeBuild compilazione a una pipeline \(CodePipeline console\)](#)
- [Aggiungere un'azione CodeBuild di test a una pipeline \(CodePipeline console\)](#)

Prerequisiti

1. Rispondere alle domande in [Pianifica una compilazione](#).
2. Se utilizzi un utente per accedere CodePipeline anziché un account AWS root o un utente amministratore, allega la policy gestita denominata `AWSCodePipelineFullAccess` all'utente (o al gruppo IAM a cui appartiene l'utente). L'utilizzo di un account AWS root non è consigliato. Questa policy concede all'utente l'autorizzazione per creare la pipeline in CodePipeline. Per ulteriori informazioni, vedere [Allegare politiche gestite](#) nella Guida per l'utente.

Note

L'entità IAM che allega la policy all'utente (o al gruppo IAM a cui appartiene l'utente) deve avere l'autorizzazione in IAM per allegare le policy. Per ulteriori informazioni, consulta [Delegare le autorizzazioni per amministrare utenti, gruppi e credenziali IAM](#) nella Guida per l'utente.

3. Crea un ruolo CodePipeline di servizio, se non ne hai già uno disponibile nel tuo account. AWS CodePipeline utilizza questo ruolo di servizio per interagire con altri AWS servizi, tra cui AWS CodeBuild, per tuo conto. Ad esempio, per utilizzare il comando IAM AWS CLI per creare un ruolo CodePipeline di servizio, esegui il `create-role` comando IAM:

Per Linux, macOS o Unix:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-  
Service-Role --assume-role-policy-document '{"Version":  
  "2012-10-17",      "Statement":{"Effect":"Allow","Principal":  
{"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}}'
```

Per Windows:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-  
role-policy-document "{\"Version\":\"2012-10-17\",      \"Statement\":{\"Effect  
\": \"Allow\", \"Principal\": {\"Service\": \"codepipeline.amazonaws.com\"}, \"Action\":  
\"sts:AssumeRole\"}}"
```

 Note

L'entità IAM che crea questo ruolo CodePipeline di servizio deve avere l'autorizzazione in IAM per creare ruoli di servizio.

4. Dopo aver creato un ruolo di CodePipeline servizio o averne identificato uno esistente, è necessario aggiungere la politica predefinita CodePipeline del ruolo di servizio al ruolo di servizio come descritto in [Rivedi la politica predefinita del ruolo di CodePipeline servizio](#) nella Guida per AWS CodePipeline l'utente, se non fa già parte della politica per il ruolo.

 Note

L'entità IAM che aggiunge questa policy sui ruoli di CodePipeline servizio deve disporre dell'autorizzazione in IAM per aggiungere le politiche dei ruoli di servizio ai ruoli di servizio.

5. Crea e carica il codice sorgente in un tipo di repository supportato da CodeBuild e CodePipeline, ad esempio CodeCommit, Amazon S3, Bitbucket o. GitHub Il codice sorgente deve contenere un file di specifica di compilazione, ma sarà possibile dichiararne uno più avanti, al momento di definire un progetto di compilazione. Per ulteriori informazioni, consulta [Riferimento per buildspec](#).

⚠ Important

Se si prevede di utilizzare la pipeline per distribuire il codice sorgente, l'artefatto di output della compilazione deve essere compatibile con il sistema di distribuzione da utilizzare.

- Per OpsWorks, consulta il [codice sorgente dell'applicazione](#) e l'[utilizzo CodePipeline con OpsWorks](#) nella Guida per l'utente.OpsWorks

Crea una pipeline che utilizza CodeBuild (CodePipeline console)

Utilizza la procedura seguente per creare una pipeline da utilizzare CodeBuild per creare e distribuire il codice sorgente.

Per creare una pipeline che testa solo il codice sorgente:

- Utilizza la procedura seguente per creare la pipeline ed eliminare le fasi Build e Beta dalla pipeline. Utilizza la procedura [Aggiungere un'azione CodeBuild di test a una pipeline \(CodePipeline console\)](#) in questo argomento per aggiungere un'operazione test alla pipeline che utilizza CodeBuild.
- Utilizza una delle altre procedure in questo argomento per creare la pipeline e utilizza la procedura [Aggiungere un'azione CodeBuild di test a una pipeline \(CodePipeline console\)](#) in questo argomento per aggiungere un'operazione test alla pipeline che utilizza CodeBuild.

Per utilizzare la procedura guidata di creazione di una pipeline CodePipeline per creare una pipeline che utilizza CodeBuild

1. Accedi a utilizzando: Console di gestione AWS

- Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, vedere [L'utente root dell'account](#) nella guida per l'utente.
- Un utente amministratore del tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a utilizzare il seguente set minimo di azioni:

```
codepipeline:*  
iam:ListRoles
```

```
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Apri la AWS CodePipeline console su <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Nel selettore AWS Regione, scegli la regione in cui si trovano le risorse del AWS tuo progetto di build. AWS Questa deve essere una AWS regione in cui CodeBuild è supportata. Per ulteriori informazioni, consulta [AWS CodeBuild](#) nella Riferimenti generali di Amazon Web Services.
4. Creare una pipeline. Se viene visualizzata una pagina di CodePipeline informazioni, scegliete Crea pipeline. Se viene visualizzata una pagina Pipelines (Pipeline), selezionare Create pipeline (Crea pipeline).
5. Nella pagina Step 1: Choose pipeline settings (Fase 1: scegli le impostazioni della pipeline), per Pipeline name (Nome pipeline), inserire un nome per la pipeline, ad esempio **CodeBuildDemoPipeline**. Se si seleziona un nome diverso, assicurarsi di utilizzarlo in tutta questa procedura.
6. In Role name (Nome ruolo), eseguire una delle seguenti operazioni:

Selezionare New service role (Nuovo ruolo di servizio), quindi, in Role Name (Nome ruolo), immettere il nome del nuovo ruolo del servizio.

Selezionare Existing service role (Ruolo di servizio esistente), quindi selezionare il ruolo del servizio CodePipeline creato o identificato come parte dei prerequisiti di questo argomento.

7. In Artifact store (Store artefatto), eseguire una delle seguenti operazioni:

- Scegli Posizione predefinita per utilizzare l'archivio di artefatti predefinito, ad esempio il bucket di artefatti S3 designato come predefinito, per la tua pipeline nella regione che hai selezionato per la AWS tua pipeline.
- Scegli Posizione personalizzata se hai già creato un archivio di artefatti, ad esempio un bucket di artefatti S3, nella stessa regione della pipeline. AWS

 Note

Non si tratta del bucket di origine per il codice sorgente della pipeline, ma dell'archivio artefatti per la pipeline. È necessario un archivio di artefatti separato, ad esempio un bucket S3, per ogni pipeline, nella stessa regione della pipeline. AWS

8. Scegli Next (Successivo).

9. Nella pagina Step 2: Add source stage (Fase 2: aggiungere fase di origine), per Source provider (Provider origine), procedere come segue:

- Se il codice sorgente è archiviato in un bucket S3, scegli Amazon S3. Per Bucket, selezionare il bucket S3 contenente il codice sorgente. In S3 object key (Chiave oggetto S3), digitare il nome del file contenente il codice sorgente (ad esempio *file-name*.zip). Scegli Next (Successivo).
- Se il codice sorgente è archiviato in un AWS CodeCommit repository, scegli. CodeCommit Per Repository name (Nome archivio), selezionare il nome del repository che contiene il codice sorgente. Per Branch name (Nome ramo), selezionare il nome del ramo contenente la versione del codice sorgente che si desidera creare. Scegli Next (Successivo).
- Se il codice sorgente è archiviato in un GitHub repository, scegli. GitHub Scegli Connect a GitHub e segui le istruzioni per l'autenticazione con GitHub. Per Repository (Archivio), selezionare il nome dell'archivio che contiene il codice sorgente. Per Branch (Ramo), selezionare il nome del ramo contenente la versione del codice sorgente che si desidera creare.

Scegli Next (Successivo).

10. Nella pagina Step 3: Add build stage (Fase 3: aggiungere fase compilazione) per Build provider (Provider compilazione), selezionare CodeBuild.

11. Se hai già un progetto di compilazione che desideri utilizzare, per Nome progetto, scegli il nome del progetto di compilazione e vai al passaggio successivo di questa procedura.

Se devi creare un nuovo progetto di CodeBuild compilazione, segui le istruzioni riportate [Creare un progetto di compilazione \(console\)](#) e torna a questa procedura.

Se scegliete un progetto di compilazione esistente, è necessario che le impostazioni degli artefatti di output di compilazione siano già definite (anche se le CodePipeline sostituisce). Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).

 Important

Se abiliti i webhook per un CodeBuild progetto e il progetto viene utilizzato come fase di compilazione CodePipeline, vengono create due build identiche per ogni commit. Una compilazione viene attivata tramite webhook e una tramite CodePipeline. Poiché la fatturazione avviene per compilazione, saranno addebitate entrambe le compilazioni. Pertanto, se lo stai utilizzando CodePipeline, ti consigliamo di disabilitare i webhook in CodeBuild. Nella console AWS CodeBuild, deselezionare la casella Webhook. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).

12. Nella pagina Step 4: Add deploy stage (Fase 4: aggiungi fase di distribuzione), procedere in uno dei seguenti modi:
 - Se non si desidera distribuire l'artefatto di output di compilazione, selezionare Skip (Salta) e confermare la scelta quando richiesto.
 - Se si desidera distribuire l'artefatto di output di compilazione, per Deploy provider (Provider distribuzione), selezionare un provider di distribuzione e specificare le impostazioni quando richiesto.

Scegli Next (Successivo).

13. Nella pagina Review (Revisione), verificare le scelte, quindi selezionare Create pipeline (Crea pipeline).
14. Dopo che la pipeline viene eseguita con successo, è possibile ottenere l'artefatto di output di compilazione. Con la pipeline visualizzata nella CodePipeline console, nell'azione Crea, scegliete il tooltip. Prendete nota del valore di Output artifact (ad esempio,). MyAppBuild

 Note

Puoi anche ottenere l'artefatto di output della build scegliendo il link Crea artefatti nella pagina dei dettagli della build nella console. CodeBuild Per accedere a questa pagina, ignorare le fasi rimanenti in questa procedura e consultare [Visualizzazione dei dettagli delle compilazioni \(console\)](#).

15. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
16. Nell'elenco dei bucket, aprire il bucket utilizzato dalla pipeline. Il nome del bucket deve seguire il formato `codepipeline-region-ID-random-number`. Puoi usare AWS CLI per eseguire il CodePipeline get-pipeline comando per ottenere il nome del bucket, `my-pipeline-name` dov'è il nome visualizzato della tua pipeline:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Nell'output, l'oggetto pipeline contiene un oggetto `artifactStore`, che contiene un valore `location` con il nome del bucket.

17. Aprire la cartella che corrisponde al nome della pipeline (in base alla lunghezza del nome della pipeline, il nome della cartella potrebbe essere troncato) e aprire la cartella che corrisponde al valore per Output artifact (Artefatto di output), annotato in precedenza.
18. Estrai i contenuti del file `.`. Se sono presenti più file in questa cartella, estrarre i contenuti del file con il timestamp Last Modified (Ultima modifica) più recente. (Potrebbe essere necessario fornire l'estensione `.zip` al file in modo che sia possibile utilizzarlo nell'utilità ZIP del sistema). L'artefatto di output della compilazione si trova nei contenuti estratti del file.
19. Se hai richiesto di CodePipeline distribuire l'artefatto di output della build, utilizza le istruzioni del provider di distribuzione per accedere all'artefatto di output della build sugli obiettivi di distribuzione.

Crea una pipeline che utilizza CodeBuild (AWS CLI)

Utilizzate la procedura seguente per creare una pipeline da utilizzare CodeBuild per creare il codice sorgente.

Per utilizzarlo AWS CLI per creare una pipeline che distribuisca il codice sorgente creato o che verifichi solo il codice sorgente, puoi adattare le istruzioni in [Modifica una pipeline \(AWS CLI\)](#) e il [riferimento alla struttura della CodePipeline pipeline](#) nella Guida per l'utente AWS CodePipeline.

1. Crea o identifica un progetto di compilazione in CodeBuild. Per ulteriori informazioni, consulta [Creazione di un progetto di compilazione](#).

⚠ Important

Il progetto di compilazione deve definire le impostazioni di artefatto di output di compilazione (anche se CodePipeline le ignora). Per ulteriori informazioni, consultare la descrizione di artifacts in [Creazione di un progetto di compilazione \(AWS CLI\)](#).

2. Assicurati di averlo configurato AWS CLI con la chiave di AWS accesso e la chiave di accesso AWS segreta che corrispondono a una delle entità IAM descritte in questo argomento. Per ulteriori informazioni, consultare l'argomento relativo a [Come configurare AWS Command Line Interface](#) nella Guida per l'utente AWS Command Line Interface.
3. Creare un file in formato JSON che rappresenta la struttura della pipeline. Nominare il file `create-pipeline.json` o simile. Ad esempio, questa struttura in formato JSON crea una pipeline con un'operazione sorgente che fa riferimento a un bucket di input S3 e un'operazione di compilazione che utilizza CodeBuild:

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



```
        {
            "name": "MyApp"
        }
    ],
    "configuration": {
        "S3Bucket": "<bucket-name>",
        "S3ObjectKey": "<source-code-file-name.zip>"
    },
    "runOrder": 1
}
]
},
{
    "name": "Build",
    "actions": [
        {
            "inputArtifacts": [
                {
                    "name": "MyApp"
                }
            ],
            "name": "Build",
            "actionTypeId": {
                "category": "Build",
                "owner": "AWS",
                "version": "1",
                "provider": "CodeBuild"
            },
            "outputArtifacts": [
                {
                    "name": "default"
                }
            ],
            "configuration": {
                "ProjectName": "<build-project-name>"
            },
            "runOrder": 1
        }
    ]
}
],
"artifactStore": {
    "type": "S3",
    "location": "<CodePipeline-internal-bucket-name>"
}
```

```
    },  
    "name": "<my-pipeline-name>",  
    "version": 1  
  }  
}
```

In questi dati in formato JSON:

- Il valore di `roleArn` deve corrispondere all'ARN del ruolo di CodePipeline servizio creato o identificato come parte dei prerequisiti.
- I valori di `S3Bucket` e `S3ObjectKey` in `configuration` presumono che il codice sorgente venga archiviato in un bucket S3. Per le impostazioni per altri tipi di repository di codice sorgente, consultare [Riferimento per la struttura della pipeline CodePipeline](#) nella Guida per l'utente di AWS CodePipeline .
- Il valore di `ProjectName` è il nome del progetto di CodeBuild compilazione creato in precedenza in questa procedura.
- Il valore `location` è il nome del bucket S3 utilizzato da questa pipeline. Per ulteriori informazioni, consultare l'argomento relativo alla [creazione di una policy per un bucket S3 da utilizzare come archivio di artefatti per CodePipeline](#) nella Guida per l'utente di AWS CodePipeline .
- Il valore di `name` è il nome di questa pipeline. Tutti i nomi di pipeline devono essere univoci per l'account.

Sebbene questi dati descrivano solo un'azione di origine e un'azione di compilazione, è possibile aggiungere azioni per attività relative al test, alla distribuzione dell'elemento di output della build, al richiamo di AWS Lambda funzioni e altro ancora. Per ulteriori informazioni, consultare l'argomento relativo al [riferimento della struttura della pipeline AWS CodePipeline](#) nella Guida per l'utente di AWS CodePipeline .

4. Passa alla cartella che contiene il file JSON, quindi esegui il CodePipeline [create-pipeline](#) comando, specificando il nome del file:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

 Note

È necessario creare la pipeline in una AWS regione in cui CodeBuild è supportata. Per ulteriori informazioni, consulta [AWS CodeBuild](#) nella Riferimenti generali di Amazon Web Services.

I dati in formato JSON vengono visualizzati nell'output e creano la pipeline. CodePipeline

5. Per ottenere informazioni sullo stato della pipeline, esegui il CodePipeline [get-pipeline-state](#) comando, specificando il nome della pipeline:

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Nell'output, cercare le informazioni che confermano la buona riuscita della compilazione. Le ellissi (...) vengono utilizzate per visualizzare i dati che sono stati omessi per brevità.

```
{
  ...
  "stageStates": [
    ...
    {
      "actionStates": [
        {
          "actionName": "CodeBuild",
          "latestExecution": {
            "status": "SUCCEEDED",
            ...
          },
          ...
        }
      ]
    }
  ]
}
```

Se si esegue il comando troppo presto, potrebbe non essere possibile visualizzare le informazioni sull'operazione di compilazione. Potrebbe essere necessario eseguire questo

comando più volte fino a quando la pipeline ha terminato l'esecuzione dell'operazione di compilazione.

6. Dopo una compilazione riuscita, seguire le istruzioni per ottenere l'artefatto di output di compilazione. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>

Note

È anche possibile ottenere l'artefatto di output di compilazione selezionando il collegamento Build artifacts (Artefatti di compilazione) nella pagina dei dettagli di compilazione correlati nella console CodeBuild. Per accedere a questa pagina, ignorare le fasi rimanenti in questa procedura e consultare [Visualizzazione dei dettagli delle compilazioni \(console\)](#).

7. Nell'elenco dei bucket, aprire il bucket utilizzato dalla pipeline. Il nome del bucket deve seguire il formato codepipeline-*<region-ID>*-*<random-number>*. Puoi ottenere il nome del bucket dal create-pipeline.json file oppure puoi eseguire il CodePipeline get-pipeline comando per ottenere il nome del bucket.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

Nell'output, l'oggetto pipeline contiene un oggetto artifactStore, che contiene un valore location con il nome del bucket.

8. Aprire la cartella corrispondente al nome della pipeline (ad esempio *<pipeline-name>*).
9. In questa cartella, aprire la cartella denominata default.
10. Estrai i contenuti del file . Se sono presenti più file in questa cartella, estrarre i contenuti del file con il timestamp Last Modified (Ultima modifica) più recente. (Potrebbe essere necessario fornire un'estensione .zip al file per poterlo utilizzare nell'utilità ZIP del sistema). L'artefatto di output della compilazione si trova nei contenuti estratti del file.

Aggiungere un'azione di CodeBuild compilazione a una pipeline (CodePipeline console)

1. Accedi a Console di gestione AWS utilizzando:
 - Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, vedere [L'utente root dell'account](#) nella guida per l'utente.

- Un utente amministratore del tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a eseguire il seguente set minimo di azioni:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Apri la CodePipeline console su <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Nel selettore della AWS regione, scegli la regione in cui si trova la tua AWS pipeline. Questa deve essere una regione in cui CodeBuild è supportata. Per ulteriori informazioni, consulta [CodeBuild](#) nella Riferimenti generali di Amazon Web Services.
4. Nella pagina Pipelines (Pipeline), selezionare il nome della pipeline.
5. Nella pagina dei dettagli della pipeline, alla voce Source (Origine), selezionare la descrizione del comando. Prendi nota del valore di Output artifact (ad esempio, MyApp).

Note

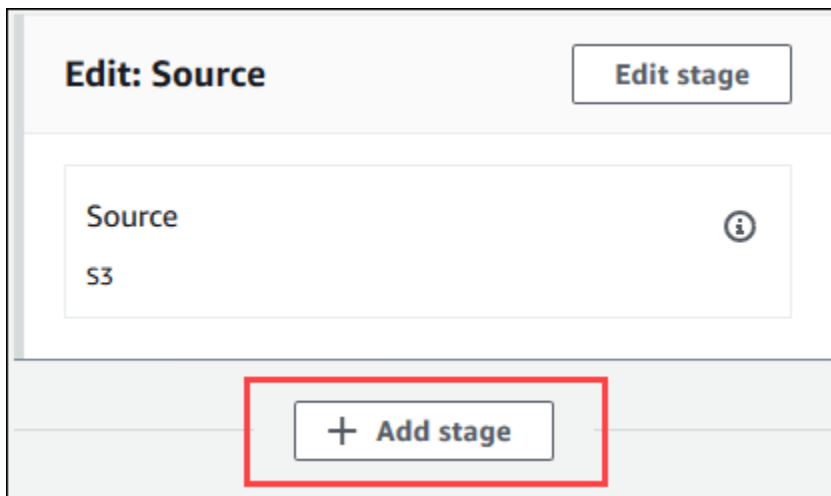
Questa procedura illustra come aggiungere un'operazione di compilazione in una fase di compilazione tra le fasi Source (Origine) e Beta. Se si desidera aggiungere l'operazione

di compilazione altrove, selezionare la descrizione del comando relativo all'operazione appena precedente al punto in cui si desidera aggiungere l'operazione di compilazione e annotare il valore per Output artifact (Artefatto di output).

6. Scegli Modifica.
7. Tra le fasi Source (Origine) e Beta, selezionare Add stage (Aggiungi fase).

Note

Questa procedura illustra come aggiungere un'operazione di compilazione tra le fasi Source (Origine) e Beta alla pipeline. Per aggiungere un'operazione di compilazione a una fase esistente, selezionare Edit stage (Modifica fase) nella fase esistente e passare alla fase 8 di questa procedura. Per aggiungere la fase di compilazione altrove, selezionare Add stage (Aggiungi fase) nella posizione desiderata.



8. Per Stage name (Nome fase), immettere il nome della fase di compilazione (ad esempio **Build**). Se si seleziona un nome diverso, utilizzarlo in tutta questa procedura.
9. All'interno della fase selezionata, selezionare Add action (Aggiungi operazione).

Note

Questa procedura illustra come aggiungere l'operazione di compilazione all'interno di una fase di compilazione. Per aggiungere l'operazione di compilazione altrove, selezionare Add action (Aggiungi operazione) nella posizione desiderata. Potrebbe prima essere

necessario selezionare Edit stage (Modifica fase) nella fase esistente in cui si desidera aggiungere l'operazione di compilazione.

10. In Edit action (Modifica operazione), in Action name (Nome operazione), immettere un nome per l'operazione (ad esempio **CodeBuild**). Se si seleziona un nome diverso, utilizzarlo in tutta questa procedura.
11. Per Action provider (Provider operazione), selezionare CodeBuild.
12. Se hai già un progetto di compilazione che desideri utilizzare, per Nome progetto, scegli il nome del progetto di compilazione e vai al passaggio successivo di questa procedura.

Se devi creare un nuovo progetto di CodeBuild compilazione, segui le istruzioni riportate [Creare un progetto di compilazione \(console\)](#) e torna a questa procedura.

Se scegliete un progetto di compilazione esistente, è necessario che le impostazioni degli artefatti di output di compilazione siano già definite (anche se le CodePipeline sostituisce). Per ulteriori informazioni, consultare la descrizione di Artifacts (Artefatti) in [Creare un progetto di compilazione \(console\)](#) o [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).

Important

Se abiliti i webhook per un CodeBuild progetto e il progetto viene utilizzato come fase di compilazione CodePipeline, vengono create due build identiche per ogni commit. Una compilazione viene attivata tramite webhook e una tramite CodePipeline. Poiché la fatturazione avviene per compilazione, saranno addebitate entrambe le compilazioni. Pertanto, se lo stai utilizzando CodePipeline, ti consigliamo di disabilitare i webhook in CodeBuild. Nella CodeBuild console, deseleziona la casella Webhook. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#)

13. In Input artifacts (Artefatti di input), selezionare l'artefatto di output annotato in precedenza in questa procedura.
14. In Output artifacts (Artefatti di output), digitare un nome per l'artefatto di output (ad esempio **MyAppBuild**).
15. Selezionare Add action (Aggiungi operazione).
16. Selezionare Save (Salva) e quindi Save (Salva) per salvare le modifiche alla pipeline.
17. Selezionare Release change (Rilascia modifica).

18. Dopo che la pipeline viene eseguita con successo, è possibile ottenere l'artefatto di output di compilazione. Con la pipeline visualizzata nella CodePipeline console, nell'azione Crea, scegliete il tooltip. Prendete nota del valore di Output artifact (ad esempio,). MyAppBuild

 Note

Puoi anche ottenere l'artefatto di output della build scegliendo il link Crea artefatti nella pagina dei dettagli della build nella console. CodeBuild Per accedere a questa pagina, consultare [Visualizzazione dei dettagli delle compilazioni \(console\)](#) e passare alla fase 31 di questa procedura.

19. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
20. Nell'elenco dei bucket, aprire il bucket utilizzato dalla pipeline. Il nome del bucket deve seguire il formato codepipeline-*region-ID-random-number*. Puoi usare AWS CLI per eseguire il CodePipeline get-pipeline comando per ottenere il nome del bucket:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Nell'output, l'oggetto pipeline contiene un oggetto artifactStore, che contiene un valore location con il nome del bucket.

21. Aprire la cartella che corrisponde al nome della pipeline (in base alla lunghezza del nome della pipeline, il nome della cartella potrebbe venire troncato) e aprire la cartella che corrisponde al valore per Output artifact (Artefatto di output), annotato in precedenza in questa procedura.
22. Estrai i contenuti del file . Se sono presenti più file in questa cartella, estrarre i contenuti del file con il timestamp Last Modified (Ultima modifica) più recente. (Potrebbe essere necessario fornire l'estensione .zip al file in modo che sia possibile utilizzarlo nell'utilità ZIP del sistema). L'artefatto di output della compilazione si trova nei contenuti estratti del file.
23. Se hai richiesto di CodePipeline distribuire l'artefatto di output della build, utilizza le istruzioni del provider di distribuzione per accedere all'artefatto di output della build sugli obiettivi di distribuzione.

Aggiungere un'azione CodeBuild di test a una pipeline (CodePipeline console)

1. Accedi a Console di gestione AWS utilizzando:

- Il tuo account AWS root. Questo non è consigliato. Per ulteriori informazioni, vedere [L'utente root dell'account](#) nella guida per l'utente.
- Un utente amministratore del tuo AWS account. Per ulteriori informazioni, consulta [Creazione del primo utente e gruppo Account AWS root](#) nella Guida per l'utente.
- Un utente del tuo AWS account con l'autorizzazione a eseguire il seguente set minimo di azioni:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Apri la CodePipeline console su <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Nel selettore della AWS regione, scegli la regione in cui si trova la tua AWS pipeline. Questa deve essere una AWS regione in cui CodeBuild è supportata. Per ulteriori informazioni, consulta [AWS CodeBuild](#) nella Riferimenti generali di Amazon Web Services.
4. Nella pagina Pipelines (Pipeline), selezionare il nome della pipeline.
5. Nella pagina dei dettagli della pipeline, alla voce Source (Origine), selezionare la descrizione del comando. Prendi nota del valore di Output artifact (ad esempio, MyApp).

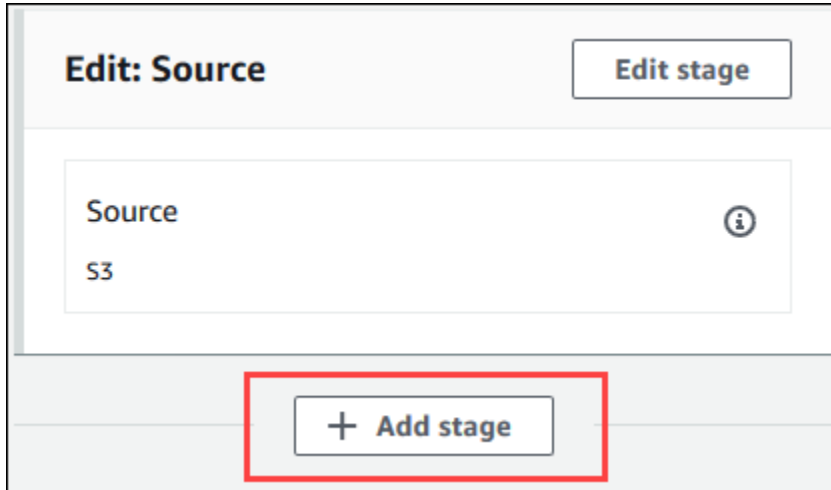
Note

Questa procedura illustra come aggiungere un'operazione di test all'interno di una fase di test tra le fasi Source (Origine) e Beta. Se si desidera aggiungere l'operazione di test altrove, posizionare il puntatore del mouse sull'operazione appena precedente e annotare il valore per Output artifact (Artefatto di output).

6. Scegli Modifica.
7. Immediatamente dopo la fase Source (Origine), selezionare Add stage (Aggiungi fase).

Note

Questa procedura illustra come aggiungere una fase di test immediatamente dopo la fase Source (Origine) alla pipeline. Per aggiungere un'operazione di test a una fase esistente, selezionare Edit stage (Modifica fase) nella fase esistente e passare alla fase 8 di questa procedura. Per aggiungere la fase di test altrove, selezionare Add stage (Aggiungi fase) nella posizione desiderata.



8. Per Stage name (Nome fase), immettere il nome della fase di test (ad esempio **Test**). Se si seleziona un nome diverso, utilizzarlo in tutta questa procedura.
9. Nella fase selezionata, selezionare Add action (Aggiungi operazione).

 Note

Questa procedura illustra come aggiungere un'operazione di test in una fase di test. Per aggiungere l'operazione di test altrove, selezionare Add action (Aggiungi operazione) nella posizione desiderata. Potrebbe prima essere necessario selezionare Edit (Modifica) nella fase esistente in cui si desidera aggiungere l'operazione di test.

10. In Edit action (Modifica operazione), in Action name (Nome operazione), immettere un nome per l'operazione (ad esempio **Test**). Se si seleziona un nome diverso, utilizzarlo in tutta questa procedura.
11. Per Action provider (Provider operazione) in Test, selezionare CodeBuild.
12. Se hai già un progetto di compilazione che desideri utilizzare, per Nome progetto, scegli il nome del progetto di compilazione e vai al passaggio successivo di questa procedura.

Se devi creare un nuovo progetto di CodeBuild compilazione, segui le istruzioni riportate [Creare un progetto di compilazione \(console\)](#) e torna a questa procedura.

 Important

Se abiliti i webhook per un CodeBuild progetto e il progetto viene utilizzato come fase di compilazione CodePipeline, vengono create due build identiche per ogni commit. Una compilazione viene attivata tramite webhook e una tramite CodePipeline. Poiché la fatturazione avviene per compilazione, saranno addebitate entrambe le compilazioni. Pertanto, se lo stai utilizzando CodePipeline, ti consigliamo di disabilitare i webhook in CodeBuild. Nella CodeBuild console, deseleziona la casella Webhook. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#)

13. Per Input artifacts (Artefatti di input), selezionare il valore per l'Output artifact (Artefatto di output) annotato in precedenza in questa procedura.
14. (Facoltativo) Se si desidera che l'operazione di test produca un artefatto di output e si configurano le specifiche di compilazione di conseguenza, in Output artifact (Artefatto di output) digitare il valore che si desidera assegnare all'artefatto di output.
15. Scegli Save (Salva).
16. Selezionare Release change (Rilascia modifica).

17. Dopo l'esecuzione della pipeline, è possibile ottenere i risultati dei test. Nella fase di test della pipeline, scegli il [collegamento ipertestuale](#) per aprire la pagina del progetto di build correlata nella console. CodeBuild
18. Nella pagina del progetto di compilazione, nella sezione Build history (Cronologia compilazione), selezionare il collegamento ipertestuale Build run (Esecuzione compilazione).
19. Nella pagina di esecuzione della compilazione, in Build logs, scegli il collegamento ipertestuale Visualizza intero log per aprire il log di build nella console Amazon CloudWatch .
20. Scorrere il log di compilazione per visualizzare i risultati dei test.

Uso AWS CodeBuild con Codecov

Codecov è uno strumento che misura la copertura del test del tuo codice. Codecov identifica quali metodi e istruzioni del codice non sono stati testati. Utilizza i risultati per determinare dove scrivere i test per migliorare la qualità del codice. Codecov è disponibile per tre dei repository di origine supportati da CodeBuild: GitHub Enterprise Server e GitHub Bitbucket. Se il progetto di compilazione utilizza GitHub Enterprise Server, è necessario utilizzare Codecov Enterprise.

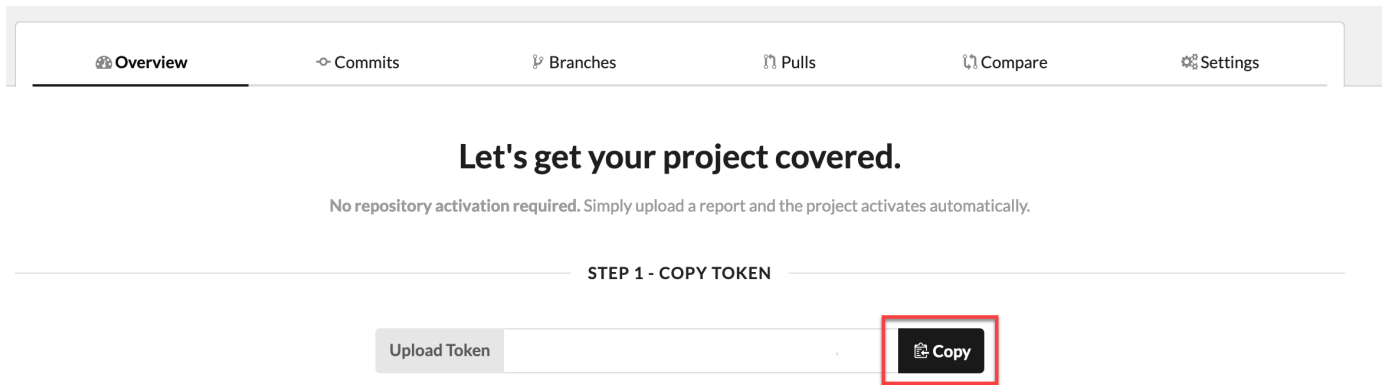
Quando esegui una build di un CodeBuild progetto integrato con Codecov, i report di Codecov che le analisi del codice nel tuo repository vengono caricate su Codecov. I log di compilazione includono un collegamento ai report. Questo esempio mostra come integrare un progetto di compilazione Python e Java con Codecov. Per l'elenco dei linguaggi supportati da Codecov, consulta la pagina con i [linguaggi supportati da Codecov](#) sul sito Web Codecov.

Integrazione di Codecov in un progetto di compilazione

Usa la seguente procedura per integrare Codecov in un progetto di compilazione.

Per integrare Codecov con il progetto di compilazione

1. Vai a <https://codecov.io/signup> e iscriviti a un repository di sorgenti GitHub o Bitbucket. Se usi GitHub Enterprise, consulta [Codecov Enterprise sul sito web di Codecov](#).
2. In Codecov, aggiungere il repository per il quale si desidera la copertura.
3. Quando vengono visualizzate le informazioni sui token, scegliere Copy (Copia).



4. Aggiungere il token copiato come variabile di ambiente denominata `CODECOV_TOKEN` al progetto di compilazione. Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(console\)](#).
5. Creare un file di testo denominato `my_script.sh` nel repository. Copiare quanto segue nel file:

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

6. Scegliere la scheda Python o Java come appropriato per il progetto di compilazione e seguire questi passaggi.

Java

1. Aggiungi il seguente JaCoCo plugin `pom.xml` nel tuo repository.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.2</version>
      <executions>
        <execution>
          <goals>
            <goal>prepare-agent</goal>
          </goals>
        </execution>
        <execution>
          <id>report</id>
          <phase>test</phase>
          <goals>
            <goal>report</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

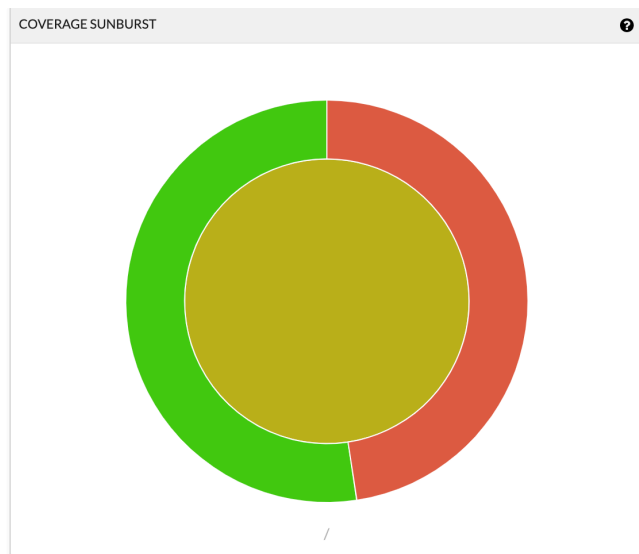


```
Bash-20200303-bc4d7e6

.[0;90m==>.[0m AWS CodeBuild detected.
... The full list of Codecov log entries has been omitted for brevity ...
.
.[0;32m->.[0m View reports at .[0;36mhttps://codecov.io/github/user/test\_py/commit/commit-id.[0m

[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

I report sono come segue:



Files					Coverage	
code.py	10	7	0	3	70.00%	
tests.py	11	11	0	0	100.00%	
Project Totals (2 files)	21	18	0	3	85.71%	

Use AWS CodeBuild con Jenkins

Puoi usare il plugin Jenkins per l'integrazione CodeBuild con i tuoi job AWS CodeBuild di build Jenkins. Invece di inviare le attività di compilazione ai relativi nodi Jenkins, puoi utilizzare il plugin per inviarle ad CodeBuild. In questo modo non è necessario eseguire il provisioning, configurare e gestire nodi di compilazione Jenkins.

Argomenti

- [Configura Jenkins](#)
- [Installazione del plug-in](#)
- [Usa il plugin](#)

Configura Jenkins

Per informazioni sulla configurazione di Jenkins con il AWS CodeBuild plugin e per scaricare il codice sorgente del plugin, consulta. <https://github.com/aws-labs/aws-codebuild-jenkins-plugin>

Installazione del plug-in

Se hai già un server Jenkins impostato e vuoi solo installare il plugin AWS CodeBuild , accedi alla tua istanza Jenkins e quindi, nel Plugin Manager, cerca **CodeBuild Plugin for Jenkins**.

Usa il plugin

Da utilizzare AWS CodeBuild con sorgenti esterne a un VPC

1. Crea un progetto nella CodeBuild console. Per ulteriori informazioni, consulta [Creare un progetto di compilazione \(console\)](#).
 - Scegli la AWS regione in cui desideri eseguire la build.
 - (Facoltativo) Imposta la configurazione Amazon VPC per consentire al container di CodeBuild build di accedere alle risorse nel tuo VPC.
 - Annotare il nome del progetto. Servirà nella fase 3.
 - (Facoltativo) Se il tuo repository di origine non è supportato nativamente da CodeBuild, puoi impostare Amazon S3 come tipo di sorgente di input per il tuo progetto.
2. In IAMconsole, crea un utente da utilizzare con il plug-in Jenkins.
 - Durante la creazione delle credenziali per l'utente, selezionare Programmatic Access (Accesso programmatico).
 - Creare una policy simile alla seguente, quindi collegarla all'utente.

JSON

```
{  
  "Version": "2012-10-17",
```



```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Resource": [  
      "arn:aws:logs:us-east-1:111122223333:log-group:/aws/  
codebuild/{{projectName}}:*"  
    ],  
    "Action": [  
      "logs:GetLogEvents"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Resource": [  
      "arn:aws:s3:::{{inputBucket}}"  
    ],  
    "Action": [  
      "s3:GetBucketVersioning"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Resource": [  
      "arn:aws:s3:::{{inputBucket}}/{{inputObject}}"  
    ],  
    "Action": [  
      "s3:PutObject"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Resource": [  
      "arn:aws:s3:::{{outputBucket}}/*"  
    ],  
    "Action": [  
      "s3:GetObject"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Resource": [  
      "arn:aws:codebuild:us-east-1:111122223333:project/  
{{projectName}}"  
    ],  
  }  
]
```

```
        "Action": [  
            "codebuild:StartBuild",  
            "codebuild:BatchGetBuilds",  
            "codebuild:BatchGetProjects"  
        ]  
    }  
]  
}
```

3. Creare un progetto freestyle in Jenkins.

- Nella pagina Configura, scegli Aggiungi fase di compilazione, quindi scegli Esegui build on CodeBuild.
- Configurare la fase di compilazione.
 - Fornire valori per Region (Regione), Credentials (Credenziali) e Project Name (Nome progetto).
 - Selezionare Use Project source (Utilizza origine progetto).
 - Salvare la configurazione ed eseguire una compilazione da Jenkins.

4. In Source Code Management (Gestione codice di origine), selezionare la modalità di recupero dell'origine. Potrebbe essere necessario installare il GitHub plug-in (o il plug-in Jenkins per il provider del repository di origine) sul server Jenkins.

- Nella pagina Configura, scegli Aggiungi fase di compilazione, quindi scegli Esegui build on. AWS CodeBuild
- Configurare la fase di compilazione.
 - Fornire valori per Region (Regione), Credentials (Credenziali) e Project Name (Nome progetto).
 - Selezionare Use Jenkins source (Utilizza origine Jenkins).
 - Salvare la configurazione ed eseguire una compilazione da Jenkins.

Per utilizzare il AWS CodeBuild plug-in con il plug-in Jenkins pipeline

- Nella pagina del progetto della pipeline Jenkins, usa il generatore di snippet per generare uno script di pipeline da aggiungere CodeBuild come passaggio alla pipeline. Lo script generato dovrebbe essere simile al seguente:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',  
sourceControlType: 'jenkins'
```

Utilizzo AWS CodeBuild con applicazioni serverless

Il AWS Serverless Application Model (AWS SAM) è un framework open source per la creazione di applicazioni serverless. Per ulteriori informazioni, consulta l'archivio dei [modelli di applicazioni AWS serverless](#) su GitHub.

È possibile utilizzarlo AWS CodeBuild per impacchettare e distribuire applicazioni serverless che seguono lo standard AWS SAM. Per la fase di distribuzione, CodeBuild può usare AWS CloudFormation. Per automatizzare la creazione e la distribuzione di applicazioni serverless con CodeBuild e CloudFormation, puoi usare AWS CodePipeline.

Per ulteriori informazioni, consulta [Deploying Serverless Applications](#) nella Developer Guide AWS Serverless Application Model.

Risorse correlate

- Per informazioni su come iniziare AWS CodeBuild, consulta [Guida introduttiva all' AWS CodeBuild utilizzo della console](#).
- Per informazioni sulla risoluzione dei problemi in CodeBuild, vedere [Risoluzione dei problemi AWS CodeBuild](#).
- Per informazioni sulle quote in CodeBuild, vedere [Quote per AWS CodeBuild](#).

Avvisi di terze parti AWS CodeBuild per Windows

Quando si utilizzano le build CodeBuild per Windows, è possibile utilizzare alcuni pacchetti e moduli di terze parti per consentire l'esecuzione dell'applicazione integrata sui sistemi operativi Microsoft Windows e l'interazione con alcuni prodotti di terze parti. Nell'elenco seguente sono riportate le condizioni legali di terze parti applicabili che regolano l'utilizzo dei pacchetti e moduli di terze parti specificati.

Argomenti

- [1\) immagine Docker di base: windowsservercore](#)

- [2\) immagine Docker basata su Windows-choco](#)
- [3\) immagine Docker basata su Windows-git --version 2.16.2](#)
- [4\) immagine Docker basata su Windows— --versione 15.0.26320.2 microsoft-build-tools](#)
- [5\) Immagine Docker basata su Windows: nuget.commandline - versione 4.5.1](#)
- [7\) immagine Docker basata su Windows: netfx-4.6.2-devpack](#)
- [8\) immagine Docker basata su Windows: visualfsharp tools, v 4.0](#)
- [9\) immagine Docker basata su Windows— -4.6 netfx-pcl-reference-assemblies](#)
- [10\) Immagine Docker basata su Windows: visualcppbuildtools v 14.0.25420.1](#)
- [11\) Immagine Docker basata su Windows: 3-ondemand-package.cab microsoft-windows-netfx](#)
- [12\) immagine Docker basata su Windows: dotnet-sdk](#)

1) immagine Docker di base: windowsservercore

(le condizioni di licenza sono disponibili all'indirizzo:) https://hub.docker.com/_/microsoft-windows-servercore

Licenza: se si richiede e si usa questa immagine contenitore del sistema operativo per i contenitori di Windows, si accetta, comprende e fornisce il consenso all'integrazione delle condizioni di licenza seguenti:

INTEGRAZIONE ALLE CONDIZIONI DI LICENZA DEL SOFTWARE MICROSOFT

IMMAGINE CONTENITORE DEL SISTEMA OPERATIVO

Microsoft Corporation (o, in base al luogo di residenza dell'utente, una delle consociate) (a cui si fa riferimento di seguito come "Microsoft") concede in licenza all'utente questa integrazione all'immagine contenitore del sistema operativo ("integrazione"). L'utente dispone della licenza all'uso di questa integrazione unitamente al software host del sistema operativo ("software host") esclusivamente come supporto all'esecuzione della funzionalità contenitori nel software host. Le condizioni di licenza del software host si applicano all'uso dell'integrazione ed è possibile non usarle se non si dispone di una licenza per il software host. È possibile usare l'integrazione con qualsiasi copia con licenza valida del software host.

REQUISITI DI LICENZA AGGIUNTIVI (DIRITTI AND/OR D'USO)

L'uso dell'integrazione come specificato nel paragrafo precedente può comportare la creazione o la modifica di un'immagine contenitore ("immagine contenitore") inclusa la modifica di determinati

componenti dell'integrazione. Per chiarezza, un'immagine contenitore è distinta e separata da un'immagine della macchina virtuale o dell'appliance virtuale. Ai sensi delle presenti condizioni di licenza, Microsoft concede un diritto limitato alla redistribuzione di tali componenti dell'integrazione in base alle condizioni seguenti:

- (i) è possibile usare i componenti dell'integrazione solo nell'ambito e come parte dell'immagine contenitore,
- (ii) è possibile usare tali componenti di integrazione nell'immagine contenitore purché si disponga di funzionalità significative nell'immagine contenitore che è sostanzialmente separata e distinta dall'integrazione e
- (iii) si accetti di includere queste condizioni di licenza (o condizioni analoghe richieste da Microsoft o da un host) nell'immagine contenitore per concedere in licenza l'eventuale uso dei componenti di integrazione da parte degli utenti finali.

Microsoft si riserva eventuali ulteriori diritti non espressamente concessi nel presente documento.

Usando l'integrazione, l'utente accetta le presenti condizioni. Qualora l'utente non le accetti, non potrà usare l'integrazione.

[Come parte delle Condizioni di licenza supplementari per questa immagine del sistema operativo Container per i contenitori Windows, l'utente è inoltre soggetto alle condizioni di licenza del software host di Windows Server sottostanti, disponibili all'indirizzo: <https://www.microsoft.com/en-us/useterms>.](#)

2) immagine Docker basata su Windows-choco

(i termini della licenza sono disponibili all'indirizzo:) <https://github.com/chocolatey/choco/blob/master/LICENSE>

Copyright 2011 - Present RealDimensions Software, LLC

Concesso ai sensi della Licenza Apache, versione 2.0 ("licenza"); l'utilizzo dei file è consentito unicamente in conformità alla licenza. È possibile ottenere una copia della licenza all'indirizzo

<http://www.apache.org/licenses/LICENSE-2,0>

Salvo espressamente richiesto dalla legislazione vigente o concordato per iscritto, il software distribuito ai sensi della licenza viene fornito "COSÌ COME È", SENZA GARANZIE O CONDIZIONI DI ALCUN TIPO, siano esse esplicite o implicite. Fare riferimento alla licenza per consultare i vincoli e le limitazioni della licenza per una lingua specifica.

3) immagine Docker basata su Windows-git --version 2.16.2

(i termini della licenza sono disponibili all'indirizzo: <https://chocolatey.org/packages/git/2.16.2>)

Concesso in licenza sotto GNU General Public License, versione 2, disponibile all'indirizzo: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

4) immagine Docker basata su Windows— --versione 15.0.26320.2 microsoft-build-tools

(i termini della licenza sono disponibili su <https://www.visualstudio.com/license-terms/:mt171552/>)

ESTENSIONI DI MICROSOFT VISUAL STUDIO 2015, VISUAL STUDIO SHELL e C++
REDISTRIBUTABLE

Le presenti condizioni di licenza costituiscono il contratto tra Microsoft Corporation (o, in base al luogo di residenza del licenziatario, una delle sue consociate) e il licenziatario. Le presenti condizioni si applicano al software di cui sopra. Le condizioni si applicano inoltre a qualsiasi servizio o aggiornamento di Microsoft relativo al software, a meno che questo non sia accompagnato da condizioni aggiuntive.

QUALORA IL LICENZIATARIO SI ATTENGA ALLE PRESENTI CONDIZIONI DI LICENZA,
DISPORRÀ DEI DIRITTI INDICATI DI SEGUITO.

1. DIRITTI DI INSTALLAZIONE E DI UTILIZZO. Il licenziatario potrà installare e utilizzare un numero qualsiasi di copie del software.
2. CONDIZIONI PER COMPONENTI SPECIFICI.
 - a. Utilità. [Il software può contenere alcuni elementi nell'elenco delle utilità all'indirizzo visualstudio/productinfo/2015-redistribution-vs. https://docs.microsoft.com/en-us/](https://docs.microsoft.com/en-us/productinfo/2015-redistribution-vs-https://docs.microsoft.com/en-us/) È possibile copiare e installare tali elementi, se inclusi nel software, su computer propri o su altri computer di terze parti, per eseguire il debug e distribuire le applicazioni e i database sviluppati con il software. Il licenziatario deve tenere presente che le Utilità sono progettate per un utilizzo temporaneo, che Microsoft potrà non essere in grado di applicare patch alle Utilità né di aggiornarle separatamente dal resto del software e che alcune Utilità potranno consentire ad altri di accedere ai computer su cui sono installati. Di conseguenza, il licenziatario deve

eliminare tutte le Utilità installate al termine del debug o della distribuzione di applicazioni e database. Microsoft non è responsabile dell'utilizzo delle, o dell'accesso alle, Utilità che il licenziatario installa su un computer da parte di terzi.

- b. Piattaforme Microsoft. Il software può includere componenti di Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office e Microsoft SharePoint. Tali componenti sono disciplinati da contratti specifici e dai relativi criteri di supporto dei prodotti descritti nelle condizioni di licenza disponibili nella directory di installazione del componente o contenuti nella cartella "Licenze" fornita insieme al software.
- c. Componenti di Terzi. Il software può includere componenti di terze parti con note legali separate o regolati da altri accordi, come descritto nel ThirdPartyNotices file che accompagna il software. Anche nel caso in cui tali componenti siano disciplinati da contratti specifici, si applicano le dichiarazioni di non responsabilità e le limitazioni e le esclusioni di danni riportate di seguito. Il software potrà inoltre contenere componenti concessi in licenza nell'ambito di licenze open source soggette a obbligazioni di disponibilità del codice sorgente. Le copie di tali licenze, se applicabili, sono incluse nel file. ThirdPartyNotices Il licenziatario potrà ottenere da Microsoft tale codice sorgente, se e come previsto dalle rispettive licenze open source, tramite l'invio di un ordine di denaro o di un assegno di 5,00 USD a: Source Code Compliance Team, Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Il licenziatario dovrà scrivere il codice sorgente di uno o più dei componenti elencati sotto nella riga nota pagamento:
 - Remote Tools per Visual Studio 2015;
 - Standalone Profiler per Visual Studio 2015;
 - IntelliTraceCollector per Visual Studio 2015;
 - Microsoft VC++ Redistributable 2015;
 - Multibyte MFC Library per Visual Studio 2015;
 - Microsoft Build Tools 2015;
 - Feedback Client;
 - Visual Studio 2015 Integrated Shell o
 - Visual Studio 2015 Isolated Shell.

Microsoft potrà inoltre effettuare una copia del codice sorgente disponibile all'indirizzo <http://thirdpartysource.microsoft.com>.

- 3. DATI. Microsoft potrà ricevere le informazioni sul licenziatario e sull'utilizzo del software che verranno raccolte tramite il software stesso. Microsoft potrà utilizzarle per erogare i servizi e migliorare i prodotti e i servizi offerti. Il licenziatario potrà rifiutarsi esplicitamente di fornire tali

informazioni, ma non tutte, come descritto nella documentazione del prodotto. Nel software sono disponibili anche alcune funzionalità che potranno consentire al licenziatario di raccogliere dati degli utenti delle applicazioni. Qualora il licenziatario utilizzi queste funzionalità per consentire la raccolta dei dati nelle applicazioni, dovrà conformarsi alla legge applicabile, che prevede tra l'altro la fornitura di comunicazioni appropriate agli utenti delle applicazioni. Puoi trovare ulteriori informazioni sulla raccolta e l'utilizzo dei dati nella documentazione di aiuto e nell'informativa sulla privacy all'indirizzo <https://privacy.microsoft.com/en-us/privacystatement>. L'utilizzo del software da parte del licenziatario costituisce accettazione implicita di tali criteri.

4. **AMBITO DELLA LICENZA.** Il software non viene venduto, ma è concesso in licenza. Il presente contratto concede al licenziatario solo alcuni diritti di utilizzo del software. Microsoft si riserva tutti gli altri diritti. Nel limite massimo consentito dalla legge applicabile, il licenziatario potrà utilizzare il software esclusivamente nei modi espressamente concessi nel presente contratto. Nel far ciò, il licenziatario dovrà attenersi a qualsiasi limitazione tecnica presente nel software che gli consenta di utilizzarlo solo in determinati modi. Il licenziatario non potrà
 - aggirare le limitazioni tecniche presenti nel software;
 - decodificare, decompilare o disassemblare il software né in altro modo tentare di fare ciò, fatta eccezione e solo per i casi previsti da condizioni di licenza di terzi che disciplinano l'utilizzo di alcuni componenti open source che potrebbero essere inclusi nel software;
 - rimuovere, ridurre, bloccare o modificare alcuna comunicazione di Microsoft o dei suoi fornitori che sia inclusa nel software;
 - utilizzare il software in contrasto con la legge o
 - condividere, pubblicare, noleggiare o concedere in locazione il software né fornirlo come soluzione ospitata autonoma utilizzabile da terzi.
5. **LIMITAZIONI RELATIVE ALL'ESPORTAZIONE.** Il licenziatario dovrà anche conformarsi a tutte le leggi e a tutti i regolamenti nazionali e internazionali sull'esportazione applicabili al software, che includono limitazioni su destinazioni, utenti finali e utilizzo finale. Per ulteriori informazioni sulle limitazioni relative all'esportazione, il licenziatario potrà visitare la pagina (aka.ms/exporting).
6. **SERVIZI DI SUPPORTO TECNICO.** Poiché il presente software viene fornito "com'è", non è prevista la fornitura di servizi di supporto tecnico da parte di Microsoft.
7. **INTERO ACCORDO.** Il presente accordo e le condizioni che disciplinano l'utilizzo dei supplementi, degli aggiornamenti, dei Servizi basati su Internet e dei servizi di supporto tecnico utilizzati dal licenziatario costituiscono l'intero accordo relativo al software e ai servizi di supporto tecnico.
8. **LEGGE APPLICABILE.** Qualora il software sia stato acquistato negli Stati Uniti, il presente contratto è interpretato in base alla legge dello Stato di Washington e tale legge si applica ai

reclami aventi come oggetto gli inadempimenti di tale contratto, mentre a tutti gli altri reclami si applicano le leggi dello Stato di residenza del licenziatario. Qualora il licenziatario abbia acquistato il software in qualsiasi altro Paese, si applicano le leggi di tale Paese.

9. DIRITTI DEI CONSUMATORI, VARIAZIONI REGIONALI. Con il presente contratto vengono concessi determinati diritti. Il licenziatario potrà avere altri diritti, tra cui i diritti dei consumatori, ai sensi delle leggi del Paese di residenza. Autonomamente da Microsoft, il licenziatario potrebbe, inoltre, vantare ulteriori diritti direttamente nei confronti della parte presso la quale ha acquistato il software. Il presente contratto non modifica tali altri diritti qualora le leggi del Paese di residenza del licenziatario non consentano di modificarli. Ad esempio, qualora il licenziatario abbia acquistato il software in una delle aree indicate di seguito o si applichi la legge obbligatoria del Paese, sono valide le seguenti disposizioni:

- a. Australia. Il licenziatario è soggetto alle garanzie di legge previste dalla Australian Consumer Law e nessuna disposizione contenuta nel presente contratto influisce su tali diritti.
- b. Canada. Qualora il software sia stato acquistato in Canada, il licenziatario potrà interrompere la ricezione degli aggiornamenti disattivando la funzionalità di aggiornamento automatico, disconnettendo il dispositivo da Internet (tuttavia, nell'eventualità e quando il licenziatario si riconnetterà a Internet, il software riprenderà a controllare se sono presenti aggiornamenti e a installarli) oppure disinstallando il software. Nell'eventuale documentazione del prodotto è possibile che vi sia inoltre specificato come il licenziatario potrà disattivare gli aggiornamenti per il dispositivo o il software in uso.
- c. Germania e Austria.
 - i. Garanzia. Il software validamente concesso in licenza funzionerà in sostanziale conformità a quanto descritto nel materiale Microsoft fornito con il software. Tuttavia, Microsoft non fornisce alcuna garanzia contrattuale in relazione al software concesso in licenza.
 - ii. Limitazione di Responsabilità. In caso di comportamento intenzionale, colpa grave, reclami basati sulla Legge in materia di Responsabilità Prodotto, così come in caso di morte o lesioni personali, Microsoft è responsabile in conformità alla legge imperativa. In riferimento alla precedente clausola (ii), Microsoft sarà responsabile solo di colpa lieve qualora sia inadempiente a tali obbligazioni contrattuali sostanziali, il cui adempimento facilita la debita esecuzione del presente contratto, il cui inadempimento metterebbe in pericolo lo scopo del presente contratto e alla cui conformità una parte potrà costantemente fare affidamento (le cosiddette "obbligazioni fondamentali"). In altri casi di colpa lieve Microsoft non sarà responsabile di tale colpa.

10. ESCLUSIONE DI GARANZIA. IL SOFTWARE VIENE CONCESSO IN LICENZA "COM'È".
IL LICENZIATARIO LO UTILIZZA A PROPRIO RISCHIO. MICROSOFT NON RICONOSCE

CONDIZIONI O GARANZIE ESPRESSE. NELLA MISURA CONSENTITA DALLE LEGGI LOCALI DELL'UTENTE, MICROSOFT ESCLUDE LE GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ, IDONEITÀ PER UNO SCOPO PARTICOLARE E NON VIOLAZIONE.

11 LIMITAZIONE ED ESCLUSIONE DEI DANNI. IL LICENZIATARIO PUÒ RICHIEDERE A MICROSOFT E AI SUOI FORNITORI IL SOLO RISARCIMENTO PER I DANNI DIRETTI NEL LIMITE DI CINQUE DOLLARI (USD 5). NON È POSSIBILE RECUPERARE ALTRI DANNI, INCLUSI DANNI CONSEQUENZIALI, MANCATI PROFITTI, DANNI SPECIALI, INDIRETTI O INCIDENTALI. Questa limitazione si applica (a) a qualsiasi questione relativa al software, ai servizi, al contenuto (incluso il codice) sui siti Internet o nelle applicazioni di terzi e (b) ai reclami relativi a inadempimento contrattuale, inadempimento delle garanzie o delle condizioni, responsabilità oggettiva, negligenza o altro illecito civile nella misura massima consentita dalla legge applicabile.

Tale limitazione si applica anche nel caso in cui Microsoft sia stata informata o avrebbe dovuto essere informata della possibilità del verificarsi di tali danni. La limitazione o l'esclusione di cui sopra potrebbe non essere applicabile al licenziatario in quanto l'esclusione o la limitazione di danni incidentali, consequenziali o di altro tipo potrebbe non essere consentita nel Paese di residenza del licenziatario.

ID EULA: _Update3_ _ VS2015 ShellsRedist <ENU>

5) Immagine Docker basata su Windows: nuget.commandline - versione 4.5.1

(le condizioni di licenza sono disponibili all'indirizzo: .txt) [https://github.com/NuGet/ Home/blob/dev/ LICENSE](https://github.com/NuGet/Home/blob/dev/LICENSE)

Copyright (c) .NET Foundation. Tutti i diritti riservati.

Concesso ai sensi della Licenza Apache, versione 2.0 ("licenza"); l'utilizzo dei file è consentito unicamente in conformità alla licenza. È possibile ottenere una copia della licenza all'indirizzo

<http://www.apache.org/licenses/LICENSE-2,0>

Salvo espressamente richiesto dalla legislazione vigente o concordato per iscritto, il software distribuito ai sensi della licenza viene fornito "COSÌ COME È", SENZA GARANZIE O CONDIZIONI DI ALCUN TIPO, siano esse esplicite o implicite. Fare riferimento alla licenza per consultare i vincoli e le limitazioni della licenza per una lingua specifica.

7) immagine Docker basata su Windows: netfx-4.6.2-devpack

INTEGRAZIONE ALLE CONDIZIONI DI LICENZA DEL SOFTWARE MICROSOFT

.NET FRAMEWORK E LANGUAGE PACK ASSOCIATI PER IL SISTEMA OPERATIVO MICROSOFT WINDOWS

Microsoft Corporation (o, in base al luogo di residenza del licenziatario, una delle sue consociate) concede in licenza la presente integrazione al licenziatario. Se l'utente dispone della licenza all'uso del software del sistema operativo Microsoft Windows ("software"), è possibile usare l'integrazione. Non è possibile usarla se non si dispone di una licenza per il software. È possibile usare l'integrazione con qualsiasi copia con licenza valida del software.

Le seguenti condizioni di licenza descrivono termini di utilizzo aggiuntivi per la presente integrazione. Le presenti condizioni unitamente alle condizioni di licenza per il software si applicano all'uso dell'integrazione. In caso di conflitto, si applica l'integrazione alle condizioni di licenza.

UTILIZZANDO QUESTO SUPPLEMENTO, L'UTENTE ACCETTA LE PRESENTI CONDIZIONI. SE NON LI ACCETTI, NON UTILIZZARE QUESTO SUPPLEMENTO.

Qualora il licenziatario si attenga alle presenti condizioni di licenza, disporrà dei diritti indicati di seguito.

1. CODICE DISTRIBUIBILE. Il supplemento è composto da Codice Distribuibile. Per "Codice Distribuibile" si intende del codice che l'utente è autorizzato a distribuire nei programmi che svilupperà, purché si attenga alle condizioni di seguito riportate.

a. Diritto di utilizzo e distribuzione.

- Il licenziatario potrà copiare e distribuire il formato in codice oggetto dell'integrazione.
- Distribuzione da parte di Terzi. Il licenziatario potrà autorizzare i distributori dei propri programmi a duplicare e distribuire il Codice Distribuibile come parte di tali programmi.

b. Requisiti di distribuzione. Per qualsiasi codice distribuibile distribuito, devi

- aggiungere rilevanti e significative funzionalità nei programmi;
- distribuire, per ogni file del Codice distribuibile con estensione .lib, solo i risultati dell'esecuzione di tale Codice Distribuibile attraverso un linker al programma;

- distribuire il Codice Distribuibile incluso in un programma di installazione solo nell'ambito di tale programma e senza alcuna modifica;
- far accettare ai distributori e agli utenti finali esterni condizioni che garantiscano al software almeno lo stesso livello di tutela definito nel presente contratto;
- visualizzare una valida comunicazione di copyright del licenziatario sui programmi del licenziatario stesso e
- indennizzare, manlevare e difendere Microsoft da ogni e qualsiasi domanda o azione giudiziaria, ivi incluse le spese legali, relative all'utilizzo o alla distribuzione dei programmi del licenziatario.

c. Restrizioni alla distribuzione. Non puoi

- modificare le eventuali comunicazioni relative ai copyright, ai marchi o ai brevetti riportati sul Codice Distribuibile;
- utilizzare i marchi di Microsoft nei nomi dei programmi o in modo tale da far presumere che i programmi provengano o siano garantiti o riconosciuti da Microsoft;
- distribuire Codice Distribuibile da eseguire su una piattaforma diversa dalla piattaforma Windows;
- includere Codice Distribuibile in programmi dannosi, ingannevoli o illegali oppure
- modificare o distribuire il codice sorgente di qualsiasi Codice Distribuibile in modo che qualsiasi parte di tale codice sia soggetto a una Licenza Esclusa. Per Licenza Esclusa si intende qualsiasi licenza che come condizione per l'utilizzo, la modifica o la distribuzione richieda che
 - il codice sia divulgato o distribuito nel formato in codice sorgente oppure
 - altri abbiano il diritto di modificarlo.

2. SERVIZI DI SUPPORTO TECNICO PER L'INTEGRAZIONE. Microsoft fornisce servizi di supporto per questo software come descritto all'[indirizzo www.support.microsoft.com/common/international.aspx](https://www.support.microsoft.com/common/international.aspx).

8) immagine Docker basata su Windows: visualfsharpools, v 4.0

(le condizioni di licenza sono disponibili all'indirizzo: .txt) <https://github.com/dotnet/fsharp/blob/main/License>

Copyright (c) Microsoft Corporation. Tutti i diritti riservati.

Concesso ai sensi della Licenza Apache, versione 2.0 ("licenza"); l'utilizzo dei file è consentito unicamente in conformità alla licenza. È possibile ottenere una copia della licenza all'indirizzo

<http://www.apache.org/licenses/LICENSE-2.0>

Salvo espressamente richiesto dalla legislazione vigente o concordato per iscritto, il software distribuito ai sensi della licenza viene fornito "COSÌ COME È", SENZA GARANZIE O CONDIZIONI DI ALCUN TIPO, siano esse esplicite o implicite. Fare riferimento alla licenza per consultare i vincoli e le limitazioni della licenza per una lingua specifica.

9) immagine Docker basata su Windows— -4.6 netfx-pcl-reference-assemblies

CONDIZIONI DI LICENZA DEL SOFTWARE MICROSOFT

ASSEMBLY DI RIFERIMENTO LIBRERIA DI CLASSI PORTABILE DI MICROSOFT .NET 4.6

Le presenti condizioni di licenza costituiscono il contratto tra Microsoft Corporation (o, in base al luogo di residenza del licenziatario, una delle sue consociate) e il licenziatario. Il licenziatario dovrà leggerle con attenzione. Le presenti condizioni si applicano al software di cui sopra. Le presenti condizioni si applicano inoltre a

- aggiornamenti,
- supplementi,
- servizi basati su Internet e
- servizi di supporto tecnico

forniti da Microsoft relativi al predetto software, a meno che questi non siano accompagnati da condizioni specifiche. In tal caso, tali condizioni specifiche prevarranno su quelle del presente contratto

UTILIZZANDO IL SOFTWARE, L'UTENTE ACCETTA LE PRESENTI CONDIZIONI. SE L'UTENTE NON LI ACCETTA, NON DEVE UTILIZZARE IL SOFTWARE.

QUALORA IL LICENZIATARIO SI ATTENGA ALLE PRESENTI CONDIZIONI DI LICENZA, DISPORRÀ DEI DIRITTI PERPETUI INDICATI DI SEGUITO.

1. DIRITTI DI INSTALLAZIONE E DI UTILIZZO. Il licenziatario potrà installare e utilizzare un numero qualsiasi di copie del software nei propri dispositivi per progettare, sviluppare e provare i propri programmi.

2. REQUISITI DI LICENZA AGGIUNTIVI DIRITTI AND/OR D'USO.

a. Codice Distribuibile. Il licenziatario è autorizzato a distribuire il software nei programmi dello strumento per sviluppatori che svilupperà, per consentire ai clienti dei propri programmi di sviluppare librerie portatili da utilizzare con qualsiasi dispositivo o sistema operativo, purché si attenga alle condizioni di seguito riportate.

i. Diritto di utilizzo e distribuzione. Il software è «Codice distribuibile».

- Codice Distribuibile. Il licenziatario potrà copiare e distribuire il formato in codice oggetto del software.
- Distribuzione da parte di Terzi. Il licenziatario potrà autorizzare i distributori dei propri programmi a duplicare e distribuire il Codice Distribuibile come parte di tali programmi.

ii. Requisiti di distribuzione. Per qualsiasi codice distribuibile distribuito, devi

- aggiungere rilevanti e significative funzionalità nei programmi;
- far accettare ai distributori e ai clienti condizioni che garantiscano al software almeno lo stesso livello di tutela definito nel presente contratto;
- visualizzare una valida comunicazione di copyright del licenziatario sui programmi del licenziatario stesso e
- indennizzare, manlevare e difendere Microsoft da ogni e qualsiasi domanda o azione giudiziaria, ivi incluse le spese legali, relative all'utilizzo o alla distribuzione dei programmi del licenziatario.

iii. Restrizioni alla distribuzione. Non puoi

- modificare le eventuali comunicazioni relative ai copyright, ai marchi o ai brevetti riportati sul Codice Distribuibile;
- utilizzare i marchi di Microsoft nei nomi dei programmi o in modo tale da far presumere che i programmi provengano o siano garantiti o riconosciuti da Microsoft;
- includere Codice Distribuibile in programmi dannosi, ingannevoli o illegali oppure
- modificare o distribuire il Codice Distribuibile in modo che qualsiasi parte di tale codice sia soggetto a una Licenza Esclusa. Per Licenza Esclusa si intende qualsiasi licenza che come condizione per l'utilizzo, la modifica o la distribuzione richieda che
 - il codice sia divulgato o distribuito nel formato in codice sorgente oppure

- altri abbiano il diritto di modificarlo

3. **AMBITO DELLA LICENZA.** Il software non viene venduto, ma è concesso in licenza. Il presente contratto concede al licenziatario solo alcuni diritti di utilizzo del software. Microsoft si riserva tutti gli altri diritti. Nel limite massimo consentito dalla legge applicabile, il licenziatario potrà utilizzare il software esclusivamente nei modi espressamente concessi nel presente contratto. Nel far ciò, il licenziatario dovrà attenersi a qualsiasi limitazione tecnica presente nel software che gli consenta di utilizzarlo solo in determinati modi. Il licenziatario non potrà
- aggirare le limitazioni tecniche presenti nel software;
 - decodificare, decompilare o disassemblare il software né in altro modo tentare di fare ciò, fatta eccezione e solo per i casi espressamente consentiti dalle leggi in vigore, in deroga a questa limitazione;
 - pubblicare il software per la duplicazione da parte di altri utenti o
 - noleggiare, affittare, prestare il software.
4. **COMMENTI E SUGGERIMENTI.** Il licenziatario può fornire un feedback relativo al software. Qualora il licenziatario fornisca un riscontro sul software a Microsoft, conferisce a Microsoft, senza alcun costo aggiuntivo, il diritto di utilizzare, condividere e commercializzare il riscontro in qualsiasi modo e per qualsiasi scopo. Il licenziatario conferisce anche a terzi, senza alcun costo aggiuntivo, eventuali diritti necessari ai loro prodotti, tecnologie e servizi per utilizzare o interfacciarsi con componenti specifici di un software o servizio Microsoft che includa il riscontro. Il licenziatario non fornirà a Microsoft commenti che siano soggetti a condizioni di licenza che impongano a Microsoft di concedere in licenza a terzi il suo software o la sua documentazione, a fronte dell'eventuale inclusione dei commenti nel software o nella documentazione stessa. Tali diritti avranno effetto anche dopo la scadenza del presente contratto
5. **TRASFERIMENTO A TERZI.** Il primo utente del software può trasferire il software e il presente contratto direttamente a un terzo. Prima del trasferimento, tale parte dovrà accettare l'applicazione del presente contratto al trasferimento e all'utilizzo del software. Il primo utente dovrà disinstallare il software prima di trasferirlo separatamente dal dispositivo. Il primo utente non potrà più trattenerne alcuna copia.
6. **LIMITAZIONI RELATIVE ALL'ESPORTAZIONE.** Il software è soggetto alle leggi e alle disposizioni in vigore negli Stati Uniti in materia di controllo dell'esportazione. Il licenziatario dovrà attenersi a tutte le leggi e le disposizioni locali e internazionali applicabili al software in materia di controllo delle esportazioni. Queste leggi includono limitazioni circa le destinazioni, gli utenti finali e l'utilizzo finale. Per altre informazioni, vedere www.microsoft.com/exporting.
7. **SERVIZI DI SUPPORTO TECNICO.** Poiché il presente software viene fornito "com'è", non è prevista la fornitura di servizi di supporto tecnico da parte di Microsoft.

8. **INTERO ACCORDO.** Il presente accordo e le condizioni che disciplinano l'utilizzo dei supplementi, degli aggiornamenti, dei Servizi basati su Internet e dei servizi di supporto tecnico utilizzati dal licenziatario costituiscono l'intero accordo relativo al software e a qualsiasi servizio di supporto tecnico fornito da Microsoft.
9. **LEGGE APPLICABILE.**
- a. Stati Uniti. Qualora il software sia stato acquistato negli Stati Uniti, il presente contratto è disciplinato e interpretato esclusivamente in base alla legge dello Stato di Washington e tale legge si applica ai reclami aventi ad oggetto gli inadempimenti contrattuali, indipendentemente dai principi in materia di conflitto di legge. Tutte le altre controversie, incluse quelle aventi ad oggetto la violazione della normativa a tutela dei consumatori, la violazione delle norme in materia di concorrenza sleale e l'illecito extracontrattuale, saranno disciplinate dalla legge dello Stato di residenza del licenziatario.
 - b. Al di fuori degli Stati Uniti. Qualora il licenziatario abbia acquistato il software in qualsivoglia altro Paese, il presente contratto è disciplinato dalla legge di tale Paese.
- 10 **EFFETTI GIURIDICI.** Con il presente contratto vengono concessi determinati diritti. Al licenziatario potranno essere concessi altri diritti ai sensi della legge del paese di residenza. Il licenziatario potrebbe, inoltre, avere ulteriori diritti direttamente nei confronti del soggetto da cui ha acquistato il software. Il presente contratto non modifica i diritti del licenziatario che la legge del Paese di residenza del licenziatario non consente di modificare.
- 11 **ESCLUSIONE DI GARANZIA. IL SOFTWARE VIENE CONCESSO IN LICENZA "COM'È".** IL LICENZIATARIO LO UTILIZZA A PROPRIO RISCHIO. MICROSOFT NON RICONOSCE CONDIZIONI O GARANZIE ESPRESSE. IL PRESENTE CONTRATTO NON MODIFICA EVENTUALI ULTERIORI DIRITTI DEI CONSUMATORI O GARANZIE DI LEGGE RICONOSCIUTI AL LICENZIATARIO DALLA LEGGE LOCALE. NELLA MISURA CONSENTITA DALLE LEGGI LOCALI DELL'UTENTE, MICROSOFT ESCLUDE LE GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ, IDONEITÀ PER UNO SCOPO PARTICOLARE E NON VIOLAZIONE.
- PER L'AUSTRIA, L'UTENTE DISPONE DI GARANZIE LEGALI AI SENSI DELLA LEGGE AUSTRIACA SUI CONSUMATORI E NESSUNA DELLE PRESENTI CONDIZIONI È INTESA A INCIDERE SU TALI DIRITTI.
- 12 **LIMITAZIONE ED ESCLUSIONE DI RIMEDI E DANNI.** IL LICENZIATARIO PUÒ RICHIEDERE A MICROSOFT E AI SUOI FORNITORI IL SOLO RISARCIMENTO PER I DANNI DIRETTI NEL LIMITE DI CINQUE DOLLARI (USD 5). NON È POSSIBILE RECUPERARE ALTRI DANNI, INCLUSI DANNI CONSEQUENZIALI, MANCATI PROFITTI, DANNI SPECIALI, INDIRETTI O INCIDENTALI.

Questa limitazione si applica a

- qualsiasi questione relativa al software, ai servizi, al contenuto (incluso il codice) sui siti Internet o i programmi di terzi e
- controversie relative a responsabilità contrattuale, violazioni di garanzia e delle condizioni, responsabilità oggettiva, negligenza o altra colpa nella misura massima consentita dalla legge applicabile.

Tale limitazione si applica anche nel caso in cui Microsoft sia stata informata o avrebbe dovuto essere informata della possibilità del verificarsi di tali danni. La limitazione o l'esclusione di cui sopra potrebbe non essere applicabile al licenziatario in quanto l'esclusione o la limitazione di danni incidentali, consequenziali o di altro tipo potrebbe non essere consentita nel Paese di residenza del licenziatario.

10) Immagine Docker basata su Windows: visualcppbuildtools v 14.0.25420.1

[\(i termini della licenza sono disponibili all'indirizzo: mt644918/\) https://www.visualstudio.com/license-terms/](https://www.visualstudio.com/license-terms/)

MICROSOFT VISUAL C++ BUILD TOOLS

CONDIZIONI DI LICENZA DEL SOFTWARE MICROSOFT

MICROSOFT VISUAL C++ BUILD TOOLS

Le presenti condizioni di licenza costituiscono il contratto tra Microsoft Corporation (o, in base al luogo di residenza del licenziatario, una delle sue consociate) e il licenziatario. Le presenti condizioni si applicano al software di cui sopra. Le condizioni si applicano inoltre a qualsiasi servizio o aggiornamento di Microsoft relativo al software, a meno che questo non sia accompagnato da condizioni diverse.

QUALORA IL LICENZIATARIO SI ATTENGA ALLE PRESENTI CONDIZIONI DI LICENZA, DISPORRÀ DEI DIRITTI INDICATI DI SEGUITO.

1. DIRITTI DI INSTALLAZIONE E DI UTILIZZO.

- a. Un utente potrà utilizzare copie del software per sviluppare e testare le proprie applicazioni.
2. DATI. Microsoft potrà ricevere le informazioni sul licenziatario e sull'utilizzo del software che verranno raccolte tramite il software stesso. Microsoft potrà utilizzarle per erogare i servizi e migliorare i prodotti e i servizi offerti. Il licenziatario potrà rifiutarsi esplicitamente di fornire tali informazioni, ma non tutte, come descritto nella documentazione del prodotto. Nel software sono disponibili anche alcune funzionalità che potranno consentire al licenziatario di raccogliere dati degli utenti delle applicazioni. Qualora il licenziatario utilizzi queste funzionalità per consentire la raccolta dei dati nelle applicazioni, dovrà conformarsi alla legge applicabile, che prevede tra l'altro la fornitura di comunicazioni appropriate agli utenti delle applicazioni. Ulteriori informazioni sulla raccolta e il trattamento dei dati sono disponibili nella documentazione di supporto e nell'informativa sulla privacy all'indirizzo <http://go.microsoft.com/fwlink/?LinkID=528096>. L'utilizzo del software da parte del licenziatario costituisce accettazione implicita di tali criteri.

3. CONDIZIONI PER COMPONENTI SPECIFICI.

- a. Server di Compilazione. Il software può contenere alcuni componenti Build Server elencati nei BuildServer file.TXT, and/or tutti i file elencati nell' BuildeServer elenco che segue le presenti Condizioni di licenza software Microsoft. Il licenziatario potrà duplicare e installare tali componenti, qualora siano inclusi nel software, sui propri computer di compilazione. Il licenziatario e altri all'interno dell'organizzazione potranno utilizzare i file presenti nei computer di compilazione solo per compilare, verificare e archiviare le applicazioni oppure per eseguire test di qualità o delle prestazioni durante il processo di compilazione.
- b. Piattaforme Microsoft. Il software può includere componenti di Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office e Microsoft SharePoint. Tali componenti sono disciplinati da contratti specifici e dai relativi criteri di supporto dei prodotti descritti nelle condizioni di licenza disponibili nella directory di installazione del componente o contenuti nella cartella "Licenze" fornita insieme al software.
- c. Componenti di Terzi. Il software può includere componenti di terze parti con note legali separate o regolati da altri accordi, come descritto nel ThirdPartyNotices file che accompagna il software. Anche nel caso in cui tali componenti siano disciplinati da contratti specifici, si applicano le dichiarazioni di non responsabilità e le limitazioni e le esclusioni di danni riportate di seguito.
- d. Funzionalità di gestione dei pacchetti. Il software include funzionalità di gestione pacchetti, quali NuGet, che consentono al licenziatario di scaricare altri pacchetti di Microsoft e di terzi da utilizzare con l'applicazione. Tali pacchetti sono disciplinati da licenze proprie, non dal presente contratto. Microsoft non distribuisce pacchetti di terzi né concede in licenza o fornisce alcuna garanzia al licenziatario in merito ad alcuno di tali pacchetti.

4. **AMBITO DELLA LICENZA.** Il software non viene venduto, ma è concesso in licenza. Il presente contratto concede al licenziatario solo alcuni diritti di utilizzo del software. Microsoft si riserva tutti gli altri diritti. Nel limite massimo consentito dalla legge applicabile, il licenziatario potrà utilizzare il software esclusivamente nei modi espressamente concessi nel presente contratto. Nel far ciò, il licenziatario dovrà attenersi a qualsiasi limitazione tecnica presente nel software che gli consenta di utilizzarlo solo in determinati modi. Per ulteriori informazioni, vedere <https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms> #1 - installation-and-use-rights Il licenziatario non potrà
- aggirare le limitazioni tecniche presenti nel software;
 - decodificare, decompilare o disassemblare il software né in altro modo tentare di fare ciò, fatta eccezione e solo per i casi previsti da condizioni di licenza di terzi che disciplinano l'utilizzo di alcuni componenti open source che potrebbero essere inclusi nel software;
 - rimuovere, ridurre, bloccare o modificare alcuna comunicazione di Microsoft o dei suoi fornitori;
 - utilizzare il software in contrasto con la legge o
 - condividere, pubblicare, noleggiare o concedere in locazione il software né fornirlo come soluzione ospitata autonoma utilizzabile da terzi.
5. **LIMITAZIONI RELATIVE ALL'ESPORTAZIONE.** Il licenziatario dovrà anche conformarsi a tutte le leggi e a tutti i regolamenti nazionali e internazionali sull'esportazione applicabili al software, che includono limitazioni su destinazioni, utenti finali e utilizzo finale. Per ulteriori informazioni sulle limitazioni relative all'esportazione, il licenziatario potrà visitare la pagina (aka.ms/exporting).
6. **SERVIZI DI SUPPORTO TECNICO.** Poiché il presente software viene fornito "com'è", non è prevista la fornitura di servizi di supporto tecnico da parte di Microsoft.
7. **INTERO ACCORDO.** Il presente accordo e le condizioni che disciplinano l'utilizzo dei supplementi, degli aggiornamenti, dei Servizi basati su Internet e dei servizi di supporto tecnico utilizzati dal licenziatario costituiscono l'intero accordo relativo al software e ai servizi di supporto tecnico.
8. **LEGGE APPLICABILE.** Qualora il software sia stato acquistato negli Stati Uniti, il presente contratto è interpretato in base alla legge dello Stato di Washington e tale legge si applica ai reclami aventi come oggetto gli inadempimenti di tale contratto, mentre a tutti gli altri reclami si applicano le leggi dello Stato di residenza del licenziatario. Qualora il licenziatario abbia acquistato il software in qualsiasi altro Paese, si applicano le leggi di tale Paese.
9. **DIRITTI DEI CONSUMATORI, VARIAZIONI REGIONALI.** Con il presente contratto vengono concessi determinati diritti. Il licenziatario potrà avere altri diritti, tra cui i diritti dei consumatori, ai sensi delle leggi del Paese di residenza. Autonomamente da Microsoft, il licenziatario potrebbe, inoltre, vantare ulteriori diritti direttamente nei confronti della parte presso la quale ha acquistato il

software. Il presente contratto non modifica tali altri diritti qualora le leggi del Paese di residenza del licenziatario non consentano di modificarli. Ad esempio, qualora il licenziatario abbia acquistato il software in una delle aree indicate di seguito o si applichi la legge obbligatoria del Paese, sono valide le seguenti disposizioni:

- Australia. Il licenziatario è soggetto alle garanzie di legge previste dalla Australian Consumer Law e nessuna disposizione contenuta nel presente contratto influisce su tali diritti.
- Canada. Qualora il software sia stato acquistato in Canada, il licenziatario potrà interrompere la ricezione degli aggiornamenti disattivando la funzionalità di aggiornamento automatico, disconnettendo il dispositivo da Internet (tuttavia, nell'eventualità e quando il licenziatario si riconnetterà a Internet, il software riprenderà a controllare se sono presenti aggiornamenti e a installarli) oppure disinstallando il software. Nell'eventuale documentazione del prodotto è possibile che vi sia inoltre specificato come il licenziatario potrà disattivare gli aggiornamenti per il dispositivo o il software in uso.
- Germania e Austria.
 - Garanzia. Il software validamente concesso in licenza funzionerà in sostanziale conformità a quanto descritto nel materiale Microsoft fornito con il software. Tuttavia, Microsoft non fornisce alcuna garanzia contrattuale in relazione al software concesso in licenza.
 - Limitazione di Responsabilità. In caso di comportamento intenzionale, colpa grave, reclami basati sulla Legge in materia di Responsabilità Prodotto, così come in caso di morte o lesioni personali, Microsoft è responsabile in conformità alla legge imperativa.

In riferimento alla precedente clausola (ii), Microsoft sarà responsabile solo di colpa lieve qualora sia inadempiente a tali obbligazioni contrattuali sostanziali, il cui adempimento facilita la debita esecuzione del presente contratto, il cui inadempimento metterebbe in pericolo lo scopo del presente contratto e alla cui conformità una parte potrà costantemente fare affidamento (le cosiddette "obbligazioni fondamentali"). In altri casi di colpa lieve Microsoft non sarà responsabile di tale colpa.

10 EFFETTI GIURIDICI. Con il presente contratto vengono concessi determinati diritti. Al licenziatario potranno essere concessi altri diritti ai sensi della legge dello stato o Paese di residenza. Il presente contratto non modifica i diritti del licenziatario che la legge dello stato o Paese di residenza del licenziatario non consente di modificare. Senza limitazione di quanto precede, per l'Australia, IL LICENZIATARIO È SOGGETTO ALLE GARANZIE DI LEGGE PREVISTE AI SENSI DELLA LEGGE AUSTRALIANA A TUTELA DEI CONSUMATORI (AUSTRALIAN CONSUMER LAW) E NESSUNA DISPOSIZIONE CONTENUTA NELLE PRESENTI CONDIZIONI INFLUISCE SU TALI DIRITTI

- 11 **ESCLUSIONE DI GARANZIA.** IL SOFTWARE VIENE CONCESSO IN LICENZA "COM'È". IL LICENZIATARIO LO UTILIZZA A PROPRIO RISCHIO. MICROSOFT NON RICONOSCE CONDIZIONI O GARANZIE ESPRESSE. NELLA MISURA CONSENTITA DALLE LEGGI LOCALI DELL'UTENTE, MICROSOFT ESCLUDE LE GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ, IDONEITÀ PER UNO SCOPO PARTICOLARE E NON VIOLAZIONE.
- 12 **LIMITAZIONE ED ESCLUSIONE DEI DANNI.** IL LICENZIATARIO PUÒ RICHIEDERE A MICROSOFT E AI SUOI FORNITORI IL SOLO RISARCIMENTO PER I DANNI DIRETTI NEL LIMITE DI CINQUE DOLLARI (USD 5). NON È POSSIBILE RECUPERARE ALTRI DANNI, INCLUSI DANNI CONSEGUENZIALI, MANCATI PROFITTI, DANNI SPECIALI, INDIRETTI O INCIDENTALI.

Questa limitazione si applica (a) a qualsiasi questione relativa al software, ai servizi, al contenuto (incluso il codice) sui siti Internet o nelle applicazioni di terzi e (b) ai reclami relativi a inadempimento contrattuale, inadempimento delle garanzie o delle condizioni, responsabilità oggettiva, negligenza o altro illecito civile nella misura massima consentita dalla legge applicabile.

Tale limitazione si applica anche nel caso in cui Microsoft sia stata informata o avrebbe dovuto essere informata della possibilità del verificarsi di tali danni. La limitazione o l'esclusione di cui sopra potrebbe non essere applicabile al licenziatario in quanto l'esclusione o la limitazione di danni incidentali, consequenziali o di altro tipo potrebbe non essere consentita nel Paese di residenza del licenziatario.

11) Immagine Docker basata su Windows: 3-ondemand-package.cab microsoft-windows-netfx

INTEGRAZIONE ALLE CONDIZIONI DI LICENZA DEL SOFTWARE MICROSOFT

MICROSOFT.NET FRAMEWORK 3.5 SP1 PER IL SISTEMA OPERATIVO MICROSOFT WINDOWS

Microsoft Corporation (o, in base al luogo di residenza del licenziatario, una delle sue consociate) concede in licenza la presente integrazione al licenziatario. Se l'utente dispone della licenza all'uso del software del sistema operativo Microsoft Windows (a cui si applica questo supplemento) (il "software"), è possibile usare il supplemento. Non è possibile usarla se non si dispone di una licenza per il software. È possibile usare una copia del supplemento con qualsiasi copia con licenza valida del software.

Le seguenti condizioni di licenza descrivono termini di utilizzo aggiuntivi per la presente integrazione. Le presenti condizioni unitamente alle condizioni di licenza per il software si applicano all'uso dell'integrazione. In caso di conflitto, si applica l'integrazione alle condizioni di licenza.

UTILIZZANDO QUESTO SUPPLEMENTO, L'UTENTE ACCETTA LE PRESENTI CONDIZIONI. SE NON LI ACCETTI, NON UTILIZZARE QUESTO SUPPLEMENTO.

Qualora il licenziatario si attenga alle presenti condizioni di licenza, disporrà dei diritti indicati di seguito.

1. SERVIZI DI SUPPORTO TECNICO PER L'INTEGRAZIONE. Microsoft fornisce servizi di supporto per questo software come descritto all'[indirizzo www.support.microsoft.com/common/international.aspx](http://www.support.microsoft.com/common/international.aspx).
2. TEST DI BENCHMARK DI MICROSOFT .NET. Il software include i componenti .NET Framework, Windows Communication Foundation, Windows Presentation Foundation e Windows Workflow Foundation del sistema operativo Windows (Componenti .NET). Il licenziatario potrà svolgere prove comparative interne dei componenti .NET e divulgarne gli esiti, purché si attenga alle condizioni previste all'indirizzo <http://go.microsoft.com/fwlink/?LinkID=66406>.

Indipendentemente da qualsiasi altro contratto in vigore tra il licenziatario e Microsoft, qualora il licenziatario divulghi gli esiti dei test di benchmark, Microsoft sarà autorizzata a comunicare gli esiti dei test di benchmark da essa condotti in relazione ai prodotti del licenziatario in concorrenza con il Componente .NET applicabile, purché si attenga alle stesse condizioni descritte all'indirizzo <http://go.microsoft.com/fwlink/?LinkID=66406>.

12) immagine Docker basata su Windows: dotnet-sdk

(disponibile in <https://github.com/dotnet/core/blob/main/LICENSE.TXT>)

Licenza MIT (MIT)

Copyright (c) Microsoft Corporation

A chiunque ottenga una copia del presente software e dei relativi file di documentazione (il «Software») viene concessa l'autorizzazione, gratuita, a utilizzare il Software senza restrizioni, inclusi, a titolo esemplificativo ma non esaustivo, i diritti di utilizzare, copiare, modificare, unire, pubblicare, distribuire, concedere in sublicenza, and/or vendere copie del Software e consentire alle persone a cui il Software è fornito di farlo, alle seguenti condizioni:

Le informazioni sul copyright riportate in precedenza e queste informazioni sull'autorizzazione devono essere incluse in tutte le copie o in parti considerevoli del Software.

IL SOFTWARE VIENE FORNITO "COM'È", SENZA GARANZIA DI ALCUN TIPO, ESPRESSA O IMPLICITA, INCLUSE A TITOLO ESEMPLIFICATIVO LE GARANZIE DI COMMERCIALIZZABILITÀ, IDONEITÀ A UN USO SPECIFICO E NON VIOLAZIONE. IN NESSUN CASO GLI AUTORI O I TITOLARI DEL COPYRIGHT SARANNO RESPONSABILI PER QUALSIASI RECLAMO, DANNO O ALTRO TIPO DI RESPONSABILITÀ, A SEGUITO DI AZIONE CONTRATTUALE, ILLECITO O ALTRO, DERIVANTE DA O IN CONNESSIONE AL SOFTWARE, AL SUO UTILIZZO O AD ALTRE OPERAZIONI CON LO STESSO.

Usa le chiavi di CodeBuild condizione come variabili del ruolo del servizio IAM per controllare l'accesso alla build

Con il CodeBuild build ARN, puoi limitare l'accesso alle risorse di compilazione utilizzando le chiavi di contesto per definire l'accesso alle risorse nel tuo ruolo di CodeBuild servizio. Infatti CodeBuild, le chiavi che possono essere utilizzate per controllare il comportamento di accesso alla build sono `codebuild:buildArn` e `codebuild:projectArn`. Con l'ARN del progetto di compilazione, puoi verificare se una chiamata alla tua risorsa proviene da un progetto di compilazione specifico. Per verificarlo, utilizza le chiavi `codebuild:buildArn` o `codebuild:projectArn` condition in una policy basata sull'identità IAM.

Per utilizzare i tasti `codebuild:buildArn` o `codebuild:projectArn` condition nella tua politica, includili come condizione in uno qualsiasi degli operatori di condizione ARN. Il valore della chiave deve essere una variabile IAM che si risolve in un ARN valido. Nella politica di esempio seguente, l'unico accesso consentito sarà al progetto di compilazione con l'ARN del progetto per la variabile `${codebuild:projectArn}` IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/${codebuild:projectArn}/*"
```

```
}  
]  
}
```

AWS CodeBuild tasti di condizione

AWS CodeBuild fornisce un set di chiavi di condizione che è possibile utilizzare nelle politiche IAM per applicare le politiche organizzative su CodeBuild risorse come progetti e flotte. Le chiavi di condizione coprono la maggior parte dei contesti di richiesta CodeBuild API, comprese le impostazioni di rete, le configurazioni delle credenziali e le restrizioni di elaborazione.

Argomenti

- [Applica le impostazioni di connettività VPC ai tuoi progetti e alle tue flotte](#)
- [Impedisci modifiche non autorizzate al progetto buildspec](#)
- [Limita i tipi di calcolo per le tue build](#)
- [Controlla le impostazioni delle variabili d'ambiente](#)
- [Usa le variabili nei nomi delle chiavi delle condizioni](#)
- [Verifica l'esistenza di attributi nelle richieste API](#)

Applica le impostazioni di connettività VPC ai tuoi progetti e alle tue flotte

Questa policy consente al chiamante di utilizzare le sottoreti e i gruppi di sicurezza selezionati VPCs durante la creazione di progetti e flotte. CodeBuild [Per ulteriori informazioni sulle chiavi di contesto multivalore, consulta Chiavi di contesto a valore singolo e chiavi di contesto multivalore.](#)

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": [  
      "codebuild:CreateProject",  
      "codebuild:CreateFleet"  
    ],  
    "Resource": "*",  
    "Condition": {
```



```

        "ForAnyValue:StringEquals": {
            "codebuild:vpcConfig.vpcId": [
                "vpc-01234567890abcdef",
                "vpc-abcdef01234567890"
            ],
            "codebuild:vpcConfig.subnets": [
                "subnet-1234abcd",
                "subnet-5678abcd"
            ],
            "codebuild:vpcConfig.securityGroupIds": [
                "sg-12345678abcdefghij",
                "sg-01234567abcdefghij"
            ]
        }
    }
}

```

Impedisci modifiche non autorizzate al progetto buildspec

Questa politica non consente al chiamante di sovrascrivere il buildspec sul campo. `buildspecOverride`

Note

La chiave `codebuild:source.buildspec condition` supporta solo l'operatore Null per verificare l'esistenza del campo API. Non valuta il contenuto del buildspec.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:StartBuild",
        "Resource": "*"
    }, {
        "Effect": "Deny",
        "Action": "codebuild:StartBuild",

```

```

        "Resource": "*",
        "Condition": {
            "Null": {
                "codebuild:source.buildspec": "false"
            }
        }
    }
}

```

Limita i tipi di calcolo per le tue build

[Questa policy consente di creare flotte che possono essere create solo con c5.large o m5.large con un tipo di istanza di calcolo.](#)

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:CreateFleet",
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "codebuild:computeConfiguration.instanceType": ["c5.large",
                "m5.large"]
            }
        }
    }]
}

```

Controlla le impostazioni delle variabili d'ambiente

Questa politica consente al chiamante di sostituire la variabile di STAGE ambiente impostando o. BETA GAMMA Inoltre, nega esplicitamente che sia stato eseguito l'override e STAGE nega l'override della PRODUCTION variabile di ambiente. MY_APP_VERSION Per le chiavi di contesto a più valori, consulta Chiavi di contesto a valore [singolo](#) e chiavi di contesto multivalore.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "codebuild:environment.environmentVariables/STAGE.value": [
            "BETA",
            "GAMMA"
          ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:StartBuild"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "codebuild:environment.environmentVariables/STAGE.value":
            "PRODUCTION"
        },
        "ForAnyValue:StringEquals": {
          "codebuild:environment.environmentVariables.name": [
            "MY_APP_VERSION"
          ]
        }
      }
    }
  ]
}
```

Usa le variabili nei nomi delle chiavi delle condizioni

È possibile utilizzare variabili nei nomi delle chiavi di condizione come `secondarySources/${sourceIdentifier}.location` e `secondaryArtifacts/${artifactIdentifier}.location`, dove è possibile specificare la [fonte](#) secondaria o l'identificatore di [artefatto](#) secondario nella policy IAM. La politica seguente consente al chiamante di creare un progetto con una posizione di origine specifica per la fonte secondaria. `mySecondSource` JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "codebuild:secondarySources/mySecondSource.location": "my-source-location"
        }
      }
    }
  ]
}
```

Verifica l'esistenza di attributi nelle richieste API

CodeBuild supporta le chiavi di condizione per verificare l'esistenza di alcuni campi nella richiesta API. La policy applica il requisito VPC durante la creazione o l'aggiornamento dei progetti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:CreateProject",
        "codebuild:UpdateProject"
      ]
    }
  ]
}
```

```
    ],  
    "Resource": "*",  
    "Condition": {  
        "Null": {  
            "codebuild:vpcConfig.vpcId": "false"  
        }  
    }  
  }  
}]  
}
```

Esempi di codice per CodeBuild l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzarlo CodeBuild con un kit di sviluppo AWS software (SDK).

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- [Esempi di base per l' CodeBuild utilizzo AWS SDKs](#)
 - [Azioni per l'utilizzo CodeBuild AWS SDKs](#)
 - [Utilizzo CreateProject con un AWS SDK o una CLI](#)
 - [Utilizzo ListBuilds con un AWS SDK o una CLI](#)
 - [Utilizzo ListProjects con un AWS SDK o una CLI](#)
 - [Utilizzo StartBuild con un AWS SDK o una CLI](#)

Esempi di base per l' CodeBuild utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS CodeBuild with. AWS SDKs

Esempi

- [Azioni per l'utilizzo CodeBuild AWS SDKs](#)
 - [Utilizzo CreateProject con un AWS SDK o una CLI](#)
 - [Utilizzo ListBuilds con un AWS SDK o una CLI](#)
 - [Utilizzo ListProjects con un AWS SDK o una CLI](#)
 - [Utilizzo StartBuild con un AWS SDK o una CLI](#)

Azioni per l'utilizzo CodeBuild AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole CodeBuild azioni con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove sono disponibili le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta la [documentazione di riferimento dell'API AWS CodeBuild](#).

Esempi

- [Utilizzo CreateProject con un AWS SDK o una CLI](#)
- [Utilizzo ListBuilds con un AWS SDK o una CLI](#)
- [Utilizzo ListProjects con un AWS SDK o una CLI](#)
- [Utilizzo StartBuild con un AWS SDK o una CLI](#)

Utilizzo **CreateProject** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateProject.

CLI

AWS CLI

Esempio 1: creare un progetto di AWS CodeBuild compilazione

L'create-projectesempio seguente crea un progetto di CodeBuild compilazione utilizzando i file sorgente di un bucket S3

```
aws codebuild create-project \
  --name "my-demo-project" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
input-bucket/my-source.zip\"}" \
  --artifacts "{\"type\": \"S3\", \"location\": \"codebuild-us-  
west-2-123456789012-output-bucket\"}" \
  --environment "{\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/  
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role"
```

Output:

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "name": "my-cli-demo-project",
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
    "lastModified": 1556839783.274,
    "badge": {
      "badgeEnabled": false
    },
    "queuedTimeoutInMinutes": 480,
    "environment": {
      "image": "aws/codebuild/standard:1.0",
      "computeType": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "imagePullCredentialsType": "CODEBUILD",
      "privilegedMode": false,
      "environmentVariables": []
    },
    "artifacts": {
      "location": "codebuild-us-west-2-123456789012-output-bucket",
      "name": "my-cli-demo-project",
      "namespaceType": "NONE",
      "type": "S3",
      "packaging": "NONE",
      "encryptionDisabled": false
    },
    "source": {
      "type": "S3",
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip",
      "insecureSsl": false
    },
    "timeoutInMinutes": 60,
    "cache": {
      "type": "NO_CACHE"
    },
    "created": 1556839783.274
  }
}
```



```
}
```

Esempio 2: creare un progetto di AWS CodeBuild compilazione utilizzando un file di input JSON per i parametri

L'create-projectesempio seguente crea un progetto di CodeBuild compilazione passando tutti i parametri richiesti in un file di input JSON. Crea il modello del file di input eseguendo il comando solo con --generate-cli-skeleton parameter.

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

Il file di input JSON create-project.json include il seguente contenuto.

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:1.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "serviceIAMRole"
}
```

Output:

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",

```

```
        "name": "message-util.zip"
      },
      "lastModified": 1472661575.244,
      "timeoutInMinutes": 60,
      "created": 1472661575.244,
      "environment": {
        "computeType": "BUILD_GENERAL1_SMALL",
        "image": "aws/codebuild/standard:1.0",
        "type": "LINUX_CONTAINER",
        "environmentVariables": []
      },
      "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
      },
      "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
      "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
    }
  }
}
```

Per ulteriori informazioni, consulta [Create a Build Project \(AWS CLI\) nella Guida](#) per l'AWS CodeBuild utente.

- Per i dettagli sull'API, consulta [CreateProject AWS CLI](#) Command Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea un progetto.

```
import {
  ArtifactsType,
  CodeBuildClient,
```

```
    ComputeType,
    CreateProjectCommand,
    EnvironmentType,
    SourceType,
  } from "@aws-sdk/client-codebuild";

// Create the AWS CodeBuild project.
export const createProject = async (
  projectName = "MyCodeBuilder",
  roleArn = "arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin",
  buildOutputBucket = "xxxx",
  githubUrl = "https://...",
) => {
  const codeBuildClient = new CodeBuildClient({});

  const response = await codeBuildClient.send(
    new CreateProjectCommand({
      artifacts: {
        // The destination of the build artifacts.
        type: ArtifactsType.S3,
        location: buildOutputBucket,
      },
      // Information about the build environment. The combination of
      "computeType" and "type" determines the
      // requirements for the environment such as CPU, memory, and disk space.
      environment: {
        // Build environment compute types.
        // https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-compute-types.html
        computeType: ComputeType.BUILD_GENERAL1_SMALL,
        // Docker image identifier.
        // See https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-available.html
        image: "aws/codebuild/standard:7.0",
        // Build environment type.
        type: EnvironmentType.LINUX_CONTAINER,
      },
      name: projectName,
      // A role ARN with permission to create a CodeBuild project, write to the
      artifact location, and write CloudWatch logs.
      serviceRole: roleArn,
      source: {
        // The type of repository that contains the source code to be built.
        type: SourceType.GITHUB,
```

```

        // The location of the repository that contains the source code to be
        built.
        location: githubUrl,
    },
    )),
);
console.log(response);
// {
//   '$metadata': {
//     httpStatusCode: 200,
//     requestId: 'b428b244-777b-49a6-a48d-5dffedced8e7',
//     extendedRequestId: undefined,
//     cfId: undefined,
//     attempts: 1,
//     totalRetryDelay: 0
//   },
//   project: {
//     arn: 'arn:aws:codebuild:us-east-1:xxxxxxxxxxxx:project/MyCodeBuilder',
//     artifacts: {
//       encryptionDisabled: false,
//       location: 'xxxxxx-xxxxxx-xxxxxx',
//       name: 'MyCodeBuilder',
//       namespaceType: 'NONE',
//       packaging: 'NONE',
//       type: 'S3'
//     },
//     badge: { badgeEnabled: false },
//     cache: { type: 'NO_CACHE' },
//     created: 2023-08-18T14:46:48.979Z,
//     encryptionKey: 'arn:aws:kms:us-east-1:xxxxxxxxxxxx:alias/aws/s3',
//     environment: {
//       computeType: 'BUILD_GENERAL1_SMALL',
//       environmentVariables: [],
//       image: 'aws/codebuild/standard:7.0',
//       imagePullCredentialsType: 'CODEBUILD',
//       privilegedMode: false,
//       type: 'LINUX_CONTAINER'
//     },
//     lastModified: 2023-08-18T14:46:48.979Z,
//     name: 'MyCodeBuilder',
//     projectVisibility: 'PRIVATE',
//     queuedTimeoutInMinutes: 480,
//     serviceRole: 'arn:aws:iam:xxxxxxxxxxxx:role/CodeBuildAdmin',
//     source: {

```

```
//      insecureSsl: false,
//      location: 'https://...',
//      reportBuildStatus: false,
//      type: 'GITHUB'
//    },
//    timeoutInMinutes: 60
//  }
// }
return response;
};
```

- Per ulteriori informazioni, consulta la [Guida per sviluppatori di AWS SDK per JavaScript](#).
- Per i dettagli sull'API, consulta la [CreateProject](#) sezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListBuilds** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare ListBuilds.

C++

SDK per C++

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
//! List the CodeBuild builds.
/*!
    \param sortType: 'SortOrderType' type.
    \param clientConfiguration: AWS client configuration.
    \return bool: Function succeeded.
*/
bool AwsDoc::CodeBuild::listBuilds(Aws::CodeBuild::Model::SortOrderType sortType,
```

```

        const Aws::Client::ClientConfiguration
&clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListBuildsRequest listBuildsRequest;
    listBuildsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Used for pagination.

    do {
        if (!nextToken.empty()) {
            listBuildsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListBuildsOutcome listBuildsOutcome =
codeBuildClient.ListBuilds(
            listBuildsRequest);

        if (listBuildsOutcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &ids =
listBuildsOutcome.GetResult().GetIds();
            if (!ids.empty()) {

                std::cout << "Information about each build:" << std::endl;
                Aws::CodeBuild::Model::BatchGetBuildsRequest getBuildsRequest;
                getBuildsRequest.SetIds(listBuildsOutcome.GetResult().GetIds());
                Aws::CodeBuild::Model::BatchGetBuildsOutcome getBuildsOutcome =
codeBuildClient.BatchGetBuilds(
                    getBuildsRequest);

                if (getBuildsOutcome.IsSuccess()) {
                    const Aws::Vector<Aws::CodeBuild::Model::Build> &builds =
getBuildsOutcome.GetResult().GetBuilds();
                    std::cout << builds.size() << " build(s) found." <<
std::endl;

                    for (auto val: builds) {
                        std::cout << val.GetId() << std::endl;
                    }
                } else {
                    std::cerr << "Error getting builds"
                        << getBuildsOutcome.GetError().GetMessage() <<
std::endl;

                    return false;
                }
            }
        }
    } while (nextToken != "");
}

```

```

        } else {
            std::cout << "No builds found." << std::endl;
        }

        // Get the next token for pagination.

        nextToken = listBuildsOutcome.GetResult().GetNextToken();
    } else {
        std::cerr << "Error listing builds"
                  << listBuildsOutcome.GetError().GetMessage()
                  << std::endl;
        return false;
    }
} while (!nextToken.

        empty()

        );

return true;
}

```

- Per i dettagli sull'API, consulta la [ListBuilds](#) sezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per ottenere un elenco di AWS CodeBuild build. IDs

L'`list-builds` esempio seguente ottiene un elenco CodeBuild IDs ordinato in ordine crescente.

```
aws codebuild list-builds --sort-order ASCENDING
```

L'output include un valore `nextToken` che indica che esiste altro output disponibile.

```

{
    "nextToken": "4AEA6u7J...The full token has been omitted for
    brevity...MzY20A==",

```

```
"ids": [  
  "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"  
  "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"  
    ... The full list of build IDs has been omitted for brevity ...  
  "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"  
]  
}
```

Esegui di nuovo questo comando e specifica il valore `nextToken` nella risposta precedente come parametro per ottenere la parte successiva dell'output. Ripeti l'operazione finché non viene più visualizzato un valore `nextToken` nella risposta.

```
aws codebuild list-builds --sort-order ASCENDING --next-  
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Parte successiva dell'output:

```
{  
  "ids": [  
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",  
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",  
      ... The full list of build IDs has been omitted for brevity ...  
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"  
  ]  
}
```

Per ulteriori informazioni, consulta [Visualizza un elenco di build IDs \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente

- Per i dettagli sull'API, consulta [ListBuilds AWS CLI](#) Command Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListProjects** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListProjects`.

C++

SDK per C++

 Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
//! List the CodeBuild projects.
/*!
    \param sortType: 'SortOrderType' type.
    \param clientConfiguration: AWS client configuration.
    \return bool: Function succeeded.
*/
bool AwsDoc::CodeBuild::listProjects(Aws::CodeBuild::Model::SortOrderType
sortType,
                                     const Aws::Client::ClientConfiguration
&clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListProjectsRequest listProjectsRequest;
    listProjectsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Next token for pagination.
    Aws::Vector<Aws::String> allProjects;

    do {
        if (!nextToken.empty()) {
            listProjectsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListProjectsOutcome outcome =
codeBuildClient.ListProjects(
    listProjectsRequest);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &projects =
outcome.GetResult().GetProjects();
            allProjects.insert(allProjects.end(), projects.begin(),
projects.end());
        }
    } while (outcome.IsSuccess() & !nextToken.empty());
}
```

```

        nextToken = outcome.GetResult().GetNextToken();
    }

    else {
        std::cerr << "Error listing projects" <<
outcome.GetError().GetMessage()
        << std::endl;
    }

    } while (!nextToken.empty());

    std::cout << allProjects.size() << " project(s) found." << std::endl;
    for (auto project: allProjects) {
        std::cout << project << std::endl;
    }

    return true;
}

```

- Per i dettagli sull'API, consulta la [ListProjects](#) sezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per ottenere un elenco dei nomi dei progetti di AWS CodeBuild build.

L'`list-projects` esempio seguente ottiene un elenco di progetti di CodeBuild compilazione ordinati per nome in ordine crescente.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

L'output include un valore `nextToken` che indica che esiste altro output disponibile.

```

{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",

```

```
        ... The full list of build project names has been omitted for
    brevity ...
        "codebuild-demo-project99"
    ]
}
```

Esegui di nuovo questo comando e specifica il valore `nextToken` della risposta precedente come parametro per ottenere la parte successiva dell'output. Ripeti l'operazione finché non viene più visualizzato un valore `nextToken` nella risposta.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
    "projects": [
        "codebuild-demo-project100",
        "codebuild-demo-project101",

        ... The full list of build project names has been omitted for brevity ...
        "codebuild-demo-project122"
    ]
}
```

Per ulteriori informazioni, consulta [Visualizza un elenco di nomi di progetti di compilazione \(AWS CLI\) nella Guida](#) per l'AWS CodeBuild utente.

- Per i dettagli sull'API, consulta [ListProjects AWS CLI](#) Command Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **StartBuild** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `StartBuild`.

C++

SDK per C++

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#!/ Start an AWS CodeBuild project build.
/*!
  \param projectName: A CodeBuild project name.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::startBuild(const Aws::String &projectName,
                                   const Aws::Client::ClientConfiguration
&clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::StartBuildRequest startBuildRequest;
    startBuildRequest.SetProjectName(projectName);

    Aws::CodeBuild::Model::StartBuildOutcome outcome =
codeBuildClient.StartBuild(
    startBuildRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully started build" << std::endl;
        std::cout << "Build ID: " << outcome.GetResult().GetBuild().GetId()
        << std::endl;
    }

    else {
        std::cerr << "Error starting build" << outcome.GetError().GetMessage()
        << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la [StartBuild](#) sezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per iniziare a eseguire la build di un progetto di AWS CodeBuild compilazione.

L'`start-build` esempio seguente avvia una compilazione per il CodeBuild progetto specificato. La compilazione sostituisce sia l'impostazione del progetto relativamente al numero di minuti di permanenza della compilazione nella coda prima del timeout sia le impostazioni relative agli artefatti del progetto.

```
aws codebuild start-build \  
  --project-name "my-demo-project" \  
  --queued-timeout-in-minutes-override 5 \  
  --artifacts-override {"\"type\": \"S3\", \"location\":  
  \"arn:aws:s3:::artifacts-override\", \"overrideArtifactName\": true"}
```

Output:

```
{  
  "build": {  
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role",  
    "buildStatus": "IN_PROGRESS",  
    "buildComplete": false,  
    "projectName": "my-demo-project",  
    "timeoutInMinutes": 60,  
    "source": {  
      "insecureSsl": false,  
      "type": "S3",  
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-  
source.zip"  
    },  
    "queuedTimeoutInMinutes": 5,  
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",  
    "currentPhase": "QUEUED",  
    "startTime": 1556905683.568,  
    "environment": {  
      "computeType": "BUILD_GENERAL1_MEDIUM",  
      "environmentVariables": [],  

```

```
    "type": "LINUX_CONTAINER",
    "privilegedMode": false,
    "image": "aws/codebuild/standard:1.0",
    "imagePullCredentialsType": "CODEBUILD"
  },
  "phases": [
    {
      "phaseStatus": "SUCCEEDED",
      "startTime": 1556905683.568,
      "phaseType": "SUBMITTED",
      "durationInSeconds": 0,
      "endTime": 1556905684.524
    },
    {
      "startTime": 1556905684.524,
      "phaseType": "QUEUED"
    }
  ],
  "logs": {
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=null;stream=null"
  },
  "artifacts": {
    "encryptionDisabled": false,
    "location": "arn:aws:s3:::artifacts-override/my-demo-project",
    "overrideArtifactName": true
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
  "initiator": "my-aws-account-name",
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
}
```

Per ulteriori informazioni, consulta [Run a Build \(AWS CLI\) nella Guida](#) per l'AWS CodeBuild utente.

- Per i dettagli sull'API, consulta [StartBuild AWS CLI Command Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Risoluzione dei problemi AWS CodeBuild

Utilizza le informazioni in questo argomento per individuare, diagnosticare e risolvere problemi. Per informazioni su come registrare e monitorare CodeBuild le build per risolvere i problemi, consulta.

[Registrazione di log e monitoraggio](#)

Argomenti

- [Apache Maven compila artefatti di riferimento dal repository errato](#)
- [I comandi di compilazione vengono eseguiti come root per impostazione predefinita](#)
- [Le compilazioni potrebbero fallire quando i nomi dei file non sono U.S. Caratteri inglesi](#)
- [Le build potrebbero non riuscire quando si ottengono i parametri da Amazon EC2 Parameter Store](#)
- [Impossibile accedere al filtro branch nella console CodeBuild](#)
- [Impossibile visualizzare l'esito positivo o negativo della compilazione](#)
- [Lo stato della build non viene segnalato al fornitore di origine](#)
- [Impossibile trovare e selezionare l'immagine di base della piattaforma Windows Server Core 2019](#)
- [I comandi precedenti nei file buildspec non vengono riconosciuti dai comandi successivi](#)
- [Errore "Access denied" \(Accesso negato\) durante il tentativo di eseguire il download della cache](#)
- [Errore "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" durante l'utilizzo di un'immagine di compilazione personalizzata](#)
- [Errore: «Il contenitore di compilazione è stato trovato morto prima di completare la compilazione. Il contenitore di compilazione è morto perché aveva esaurito la memoria o l'immagine Docker non è supportata. ErrorCode: 500»](#)
- [Errore: "Impossibile connettersi al daemon Docker" quando si esegue una build](#)
- [Errore: "non CodeBuild è autorizzato a eseguire: sts:AssumeRole" durante la creazione o l'aggiornamento di un progetto di compilazione](#)
- [Errore: «Errore nella chiamata GetBucketAcl: il proprietario del bucket è cambiato o il ruolo del servizio non è più autorizzato a chiamare s3:» GetBucketAcl](#)
- [Errore "Failed to upload artifacts: Invalid arn" \(Impossibile caricare artefatti: ARN non valido\) durante l'esecuzione di una compilazione](#)
- [Errore "Git Clone Failed: unable to access 'your-repository-URL': SSL certificate problem: self signed certificate" \(Git Clone non riuscito: impossibile accedere a "your-repository-URL": problema di certificato SSL: certificato autofirmato\)](#)

- [Errore "The bucket you are attempting to access must be addressed using the specified endpoint" \(Il bucket a cui tenti di accedere deve essere individuato tramite l'endpoint specificato\) durante l'esecuzione di una compilazione](#)
- [Errore: "This build image requires selecting at least one runtime version." \(Questa immagine di compilazione richiede la selezione di almeno una versione runtime.\)](#)
- [Errore: "QUEUED: INSUFFICIENT_SUBNET" quando una compilazione ha esito negativo in una coda di compilazione](#)
- [Errore: «Impossibile scaricare la cache: RequestError: Invio della richiesta non riuscito a causa di: x509: impossibile caricare le radici del sistema e nessuna root fornita»](#)
- [Errore: «Impossibile scaricare il certificato da S3. AccessDenied»](#)
- [Errore "Unable to Locate Credentials" \(Impossibile individuare le credenziali\)](#)
- [RequestError errore di timeout durante l'esecuzione CodeBuild in un server proxy](#)
- [La bourne shell \(sh\) deve esistere nelle immagini di compilazione](#)
- [Avvertenza: "Skipping install of runtimes. Runtime version selection is not supported by this build image" \(L'installazione dei runtime non viene eseguita. L'immagine di compilazione non supporta la selezione delle versioni dei runtime\) durante l'esecuzione di una compilazione](#)
- [Errore: «Impossibile verificare JobWorker l'identità» all'apertura della CodeBuild console](#)
- [Avvio della compilazione non riuscito](#)
- [Accesso ai GitHub metadati nelle build memorizzate nella cache locale](#)
- [AccessDenied: Il proprietario del bucket per il gruppo di report non corrisponde al proprietario del bucket S3...](#)
- [Errore: «Le tue credenziali non dispongono di uno o più ambiti di privilegi richiesti» durante la creazione di un progetto con CodeBuild CodeConnections](#)
- [Errore: «Siamo spiacenti, non è stato richiesto alcun terminale, impossibile ricevere input» durante la compilazione con il comando di installazione di Ubuntu](#)

Apache Maven compila artefatti di riferimento dal repository errato

Problema: [quando usi Maven con un ambiente di compilazione Java AWS CodeBuild fornito, Maven estrae le dipendenze di build e plugin dal repository Maven centrale sicuro all'indirizzo `https://repo1.maven.org/maven2`](#). Ciò avviene anche se il file `pom.xml` del progetto di compilazione dichiara esplicitamente altre posizioni alternative da utilizzare.

Possibile causa: gli ambienti di CodeBuild compilazione Java forniti includono un file denominato preinstallato nella directory dell'ambiente di compilazione. `settings.xml` /`root/.m2` Questo file `settings.xml` contiene le seguenti dichiarazioni, che indicano a Maven di eseguire sempre il pull delle dipendenze di compilazione e di plugin da un repository Maven centrale protetto disponibile all'indirizzo <https://repo1.maven.org/maven2>.

```
<settings>
  <activeProfiles>
    <activeProfile>securecentral</activeProfile>
  </activeProfiles>
  <profiles>
    <profile>
      <id>securecentral</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

Soluzione consigliata: procedi come segue:

1. Aggiungere un file `settings.xml` al codice sorgente.
2. In questo file `settings.xml` utilizzare il formato `settings.xml` precedente come guida per dichiarare i repository da cui si desidera che Maven esegua il pull delle dipendenze di compilazione e di plugin.

3. Nella `install` fase del progetto di compilazione, chiedi di CodeBuild copiare il `settings.xml` file nella directory dell'ambiente di compilazione. `/root/.m2` Considerare ad esempio il seguente frammento di un file `buildspec.yml` che dimostra questo comportamento.

```
version 0.2

phases:
  install:
    commands:
      - cp ./settings.xml /root/.m2/settings.xml
```

I comandi di compilazione vengono eseguiti come root per impostazione predefinita

Problema: AWS CodeBuild esegue i comandi di compilazione come utente root. Ciò avviene anche se il Dockerfile dell'immagine di compilazione correlata imposta l'istruzione `USER` su un altro utente.

Causa: per impostazione predefinita, CodeBuild esegue tutti i comandi di build come utente root.

Soluzione consigliata: nessuna.

Le compilazioni potrebbero fallire quando i nomi dei file non sono U.S. Caratteri inglesi

Problema: quando si esegue una build che utilizza file con nomi di file che contengono nomi non U.S. Caratteri inglesi (ad esempio caratteri cinesi), la compilazione fallisce.

Possibile causa: gli ambienti di compilazione forniti da AWS CodeBuild hanno le impostazioni locali predefinite impostate su `POSIX`. `POSIX` le impostazioni di localizzazione sono meno compatibili con CodeBuild i nomi di file che contengono dati non U.S. caratteri inglesi e possono causare il fallimento delle build correlate.

Soluzione consigliata: aggiungi i seguenti comandi alla sezione `pre_build` del file `buildspec`. Questi comandi fanno sì che l'ambiente di compilazione utilizzi UTF-8 in inglese americano per le impostazioni di localizzazione, che è più compatibile con CodeBuild i nomi di file che contengono nomi non U.S. Caratteri inglesi.

Per gli ambienti di compilazione basati su Ubuntu:

```
pre_build:
  commands:
    - export LC_ALL="en_US.UTF-8"
    - locale-gen en_US en_US.UTF-8
    - dpkg-reconfigure -f noninteractive locales
```

Per ambienti di compilazione basati su Amazon Linux:

```
pre_build:
  commands:
    - export LC_ALL="en_US.utf8"
```

Le build potrebbero non riuscire quando si ottengono i parametri da Amazon EC2 Parameter Store

Problema: quando una build tenta di ottenere il valore di uno o più parametri memorizzati in Amazon EC2 Parameter Store, la compilazione fallisce nella `DOWNLOAD_SOURCE` fase con l'errore. `Parameter does not exist`

Possibile causa: il ruolo di servizio su cui si basa il progetto di compilazione non è autorizzato a richiamare `ssm:GetParameters` oppure il progetto di compilazione utilizza un ruolo di servizio generato AWS CodeBuild e che consente di richiamare `ssm:GetParameters`, ma i parametri hanno nomi che non iniziano con `/CodeBuild/`

Soluzioni consigliate:

- Se il ruolo di servizio non è stato generato da CodeBuild, aggiorna la sua definizione per consentire di CodeBuild richiamare `ssm:GetParameters`. La seguente dichiarazione di policy consente ad esempio di chiamare l'azione `ssm:GetParameters` per ottenere parametri con nomi che iniziano con `/CodeBuild/`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
```

```

        "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
CodeBuild/*"
    }
]
}

```

- Se il ruolo di servizio è stato generato da CodeBuild, aggiorna la sua definizione per consentire l'accesso CodeBuild ai parametri in Amazon EC2 Parameter Store con nomi diversi da quelli che iniziano con `/CodeBuild/`. La seguente dichiarazione di policy consente ad esempio di chiamare l'azione `ssm:GetParameters` per ottenere parametri con il nome specificato:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
      "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/PARAMETER_NAME"
    }
  ]
}

```

Impossibile accedere al filtro branch nella console CodeBuild

Problema: l'opzione del filtro branch non è disponibile nella console quando si crea o si aggiorna un AWS CodeBuild progetto.

Possibile causa: l'opzione di filtro dei rami è obsoleto. È stata sostituita dai gruppi di filtri di webhook, che offrono maggiore controllo sugli eventi webhook che attivano una nuova compilazione in CodeBuild.

Soluzione consigliata: per effettuare la migrazione di un filtro dei rami creato prima dell'introduzione dei filtri di webhook, creare un gruppo di filtri di webhook contenente un filtro HEAD_REF con l'espressione regolare `^refs/heads/branchName$`. Ad esempio, se l'espressione regolare del tuo filtro dei rami era `^branchName$`, l'espressione regolare aggiornata da inserire nel filtro HEAD_REF sarà `^refs/heads/branchName$`. Per ulteriori informazioni, consultare [Eventi webhook Bitbucket](#) e [Filtra gli eventi GitHub webhook \(console\)](#).

Impossibile visualizzare l'esito positivo o negativo della compilazione

Problema: non riesci a visualizzare se una compilazione ripetuta ha avuto esito positivo o negativo.

Possibile causa: l'opzione per la segnalazione dello stato della compilazione non è abilitata.

Soluzioni consigliate: abilita lo stato di creazione del report quando crei o aggiorni un CodeBuild progetto. Questa opzione indica ad CodeBuild di segnalare lo stato quando si attiva una compilazione. Per ulteriori informazioni, consulta [reportBuildStatus](#) nella documentazione di riferimento dell'API AWS CodeBuild .

Lo stato della build non viene segnalato al fornitore di origine

Problema: dopo aver consentito la segnalazione dello stato della build a un provider di origine, ad GitHub esempio Bitbucket, lo stato della build non viene aggiornato.

Possibile causa: l'utente associato al provider di origine non dispone dell'accesso in scrittura al repository.

Soluzioni consigliate: per poter segnalare lo stato della build al provider di origine, l'utente associato al provider di origine deve disporre dell'accesso in scrittura al repository. Se l'utente non dispone dell'accesso in scrittura, lo stato della build non può essere aggiornato. Per ulteriori informazioni, consulta [Accesso al provider di origine](#).

Impossibile trovare e selezionare l'immagine di base della piattaforma Windows Server Core 2019

Problema: non è possibile trovare o selezionare l'immagine di base della piattaforma Windows Server Core 2019.

Possibile causa: stai utilizzando una AWS regione che non supporta questa immagine.

Soluzioni consigliate: utilizza una delle seguenti AWS regioni in cui è supportata l'immagine di base della piattaforma Windows Server Core 2019:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (Oregon)

- Europa (Irlanda)

I comandi precedenti nei file buildspec non vengono riconosciuti dai comandi successivi

Problemi: i risultati di uno o più comandi nel file buildspec non vengono riconosciuti dai comandi successivi nello stesso file buildspec. È ad esempio possibile che un comando imposti una variabile di ambiente locale, ma che una successiva esecuzione del comando non riesca a recuperare il valore di tale variabile di ambiente locale.

Possibile causa: nella versione 0.1 del file buildspec, AWS CodeBuild esegue ogni comando in un'istanza separata della shell predefinita nell'ambiente di compilazione. Questo significa che ogni comando viene eseguito separatamente da tutti gli altri comandi. Per impostazione predefinita, non è dunque possibile eseguire un singolo comando che si basa sullo stato di qualsiasi comando precedente.

Soluzioni consigliate: ti suggeriamo di utilizzare la versione 0.2 della specifica di compilazione, che risolve questo problema. Se devi utilizzare la versione 0.1 della specifica di compilazione, ti consigliamo di usare l'operatore di concatenazione del comando shell, ad esempio `&&` in Linux, per combinare più comandi in un singolo comando. In alternativa, includi uno script della shell nel codice sorgente che contiene più comandi, quindi chiama tale script della shell da un singolo comando nel file buildspec. Per ulteriori informazioni, consultare [Shell e comandi negli ambienti di compilazione](#) e [Variabili di ambiente degli ambienti di compilazione](#).

Errore "Access denied" (Accesso negato) durante il tentativo di eseguire il download della cache

Problema: quando tenti di scaricare la cache in un progetto di compilazione con la cache abilitata, viene visualizzato l'errore `Access denied`.

Possibili cause:

- Hai appena configurato il caching come parte del progetto di compilazione.
- La cache è stata recentemente invalidata tramite l'API `InvalidateProjectCache`.
- Il ruolo di servizio utilizzato da CodeBuild non dispone di `s3:GetObject` `s3:PutObject` autorizzazioni per il bucket S3 che contiene la cache.

Soluzione consigliata: nel caso sia il primo utilizzo, è normale visualizzare questo errore subito dopo l'aggiornamento della configurazione della cache. Se l'errore persiste, verifica se il ruolo del servizio dispone delle autorizzazioni `s3:GetObject` e `s3:PutObject` per il bucket S3 con la cache. Per ulteriori informazioni, consulta [Specificare le autorizzazioni S3](#) nella Amazon S3 Developer Guide.

Errore "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" durante l'utilizzo di un'immagine di compilazione personalizzata

Problema: quando tenti di eseguire una compilazione che utilizza un'immagine di compilazione personalizzata, la compilazione ha esito negativo e viene restituito l'errore `BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE`.

Possibile causa: la dimensione complessiva non compressa dell'immagine di compilazione è maggiore dello spazio su disco disponibile del tipo di calcolo dell'ambiente di compilazione. Per controllare la dimensione dell'immagine di compilazione, utilizzare Docker per eseguire il comando `docker images REPOSITORY:TAG`. Per un elenco dello spazio su disco disponibile per tipo di calcolo, consulta [Modi e tipi di calcolo dell'ambiente di creazione](#).

Soluzione consigliata: utilizza un tipo di elaborazione più grande con più spazio su disco disponibile o riduci le dimensioni dell'immagine di build personalizzata.

Possibile causa: AWS CodeBuild non dispone dell'autorizzazione per estrarre l'immagine di build dal tuo Amazon Elastic Container Registry (Amazon ECR).

Soluzione consigliata: aggiorna le autorizzazioni nel tuo repository in Amazon ECR in modo da CodeBuild poter inserire l'immagine di build personalizzata nell'ambiente di compilazione. Per ulteriori informazioni, consulta [Esempio di Amazon ECR](#).

Possibile causa: l'immagine Amazon ECR richiesta non è disponibile nella AWS regione utilizzata dal tuo AWS account.

Soluzione consigliata: usa un'immagine Amazon ECR che si trova nella stessa AWS regione di quella utilizzata dal tuo AWS account.

Possibile causa: stai utilizzando un registro privato in un VPC che non dispone di accesso pubblico a Internet. CodeBuild non è possibile estrarre un'immagine da un indirizzo IP privato in un VPC. Per ulteriori informazioni, consulta [Registro privato con Gestione dei segreti AWS esempio per CodeBuild](#).

Soluzione consigliata: se utilizzi un registro privato in un VPC, assicurati che il VPC disponga di un accesso pubblico a Internet.

Possibile causa: se il messaggio di errore contiene "toomanyrequests" e l'immagine è ottenuta da Docker Hub, questo errore significa che è stato raggiunto il limite di pull di Docker Hub.

Soluzione consigliata: utilizza un registro privato Docker Hub o ottieni la tua immagine da Amazon ECR. Per ulteriori informazioni sull'utilizzo di un registro privato, consulta [Registro privato con Gestione dei segreti AWS esempio per CodeBuild](#). Per ulteriori informazioni sull'uso di Amazon ECR, consulta [Esempio di Amazon ECR per CodeBuild](#).

Errore: «Il contenitore di compilazione è stato trovato morto prima di completare la compilazione. Il contenitore di compilazione è morto perché aveva esaurito la memoria o l'immagine Docker non è supportata. ErrorCode: 500»

Problema: quando si tenta di utilizzare un contenitore Microsoft Windows o Linux in AWS CodeBuild, questo errore si verifica durante la fase di PROVISIONING.

Possibili cause:

- La versione del sistema operativo del contenitore non è supportata da CodeBuild.
- HTTP_PROXY, HTTPS_PROXY o entrambi sono specificati nel container.

Soluzioni consigliate:

- Per Microsoft Windows, usa un contenitore Windows con un sistema operativo contenitore (versione 10.0.14393.2125) `microsoft/windowsservercore:10.0.x` (for example, `microsoft/windowsservercore`).
- Per Linux, deseleziona le impostazioni HTTP_PROXY e HTTPS_PROXY nell'immagine Docker oppure specifica la configurazione del VPC nel progetto di compilazione.

Errore: "Impossibile connettersi al daemon Docker" quando si esegue una build

Problema: la tua compilazione ha esito negativo e ricevi un errore simile a `Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?` nel log di compilazione.

Possibile causa: la tua compilazione non è in esecuzione in modalità con privilegi.

Soluzione consigliata: per correggere questo errore, devi abilitare la modalità privilegiata e aggiornare il buildspec utilizzando le seguenti istruzioni.

Per eseguire la build in modalità privilegiata, segui questi passaggi:

1. Apri la CodeBuild console all'indirizzo <https://console.aws.amazon.com/codebuild/>.
2. Nel pannello di navigazione, scegli Crea progetti, quindi scegli il tuo progetto di compilazione.
3. In Modifica, scegliere Ambiente.
4. Scegli Additional configuration (Configurazione aggiuntiva).
5. Da Privileged, seleziona Abilita questo flag se desideri creare immagini Docker o desideri che le tue build ottengano privilegi elevati. .
6. Selezionare Update environment (Aggiorna ambiente).
7. Scegliere Avvia compilazione per ritentare la compilazione della build.

Dovrai anche avviare il demone Docker all'interno del tuo contenitore. La `install` fase delle specifiche di compilazione potrebbe essere simile a questa.

```
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Per ulteriori informazioni sui driver di storage OverlayFS a cui si fa riferimento nel file buildspec, consulta [Utilizzare il driver di storage OverlayFS](#) nel sito Web Docker.

Note

Se il sistema operativo di base è Alpine Linux, in `buildspec.yml` aggiungere l'argomento `-t` a `timeout`:

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Per ulteriori informazioni su come creare ed eseguire un'immagine Docker utilizzando, consulta. AWS CodeBuild [Docker in un esempio di immagine personalizzato per CodeBuild](#)

Errore: "non CodeBuild è autorizzato a eseguire: sts:AssumeRole" durante la creazione o l'aggiornamento di un progetto di compilazione

Problema: quando tenti di creare o aggiornare un progetto di compilazione, viene visualizzato l'errore `Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name.`

Possibili cause:

- Il AWS Security Token Service (AWS STS) è stato disattivato per la AWS regione in cui si sta tentando di creare o aggiornare il progetto di compilazione.
- Il ruolo AWS CodeBuild di servizio associato al progetto di compilazione non esiste o non dispone di autorizzazioni sufficienti per essere considerato attendibile. CodeBuild
- Il case del ruolo di AWS CodeBuild servizio associato al progetto di compilazione non corrisponde al ruolo IAM effettivo.

Soluzioni consigliate:

- Assicurati che AWS STS sia attivato per la AWS regione in cui stai tentando di creare o aggiornare il progetto di compilazione. Per ulteriori informazioni, consulta [Attivazione e disattivazione AWS STS in una AWS regione nella Guida](#) per l'utente IAM.
- Assicurati che il ruolo di CodeBuild servizio di destinazione esista nel tuo account. AWS Se non utilizzi la console, assicurati di non avere scritto in modo errato l'Amazon Resource Name (ARN) del ruolo del servizio durante la creazione o l'aggiornamento del progetto di compilazione. Tieni presente che i ruoli IAM fanno distinzione tra maiuscole e minuscole, quindi verifica che l'assegnazione delle maiuscole e minuscole del ruolo IAM sia corretta.
- Assicurati che il ruolo di CodeBuild servizio di destinazione disponga di autorizzazioni sufficienti per essere considerato attendibile. CodeBuild Per ulteriori informazioni, consulta la dichiarazione di policy per relazioni di fiducia in [Consenti CodeBuild di interagire con altri servizi AWS](#).

Errore: «Errore nella chiamata GetBucketAcl: il proprietario del bucket è cambiato o il ruolo del servizio non è più autorizzato a chiamare s3:» GetBucketAcl

Problema: quando esegui una build, viene visualizzato un errore circa la modifica della proprietà di un bucket S3 e delle autorizzazioni GetBucketAcl.

Possibile causa: hai aggiunto le `s3:GetBucketLocation` autorizzazioni `s3:GetBucketAcl` and al tuo ruolo IAM. Queste autorizzazioni garantiscono la sicurezza del bucket S3 del progetto e garantiscono che solo tu puoi accedervi. Dopo aver aggiunto queste autorizzazioni, il proprietario del bucket S3 cambia.

Soluzione consigliata: verifica di essere il proprietario del bucket S3, quindi aggiungi nuovamente le autorizzazioni al tuo ruolo IAM. Per ulteriori informazioni, consulta [Accesso sicuro ai bucket S3](#).

Errore "Failed to upload artifacts: Invalid arn" (Impossibile caricare artefatti: ARN non valido) durante l'esecuzione di una compilazione

Problema: quando esegui una compilazione, la fase della compilazione `UPLOAD_ARTIFACTS` non riesce con l'errore `Failed to upload artifacts: Invalid arn`.

Possibile causa: il bucket di output S3 (il bucket in cui AWS CodeBuild memorizza l'output della build) si trova in una AWS regione diversa da quella del progetto di build. CodeBuild

Soluzione consigliata: aggiorna le impostazioni del progetto di build in modo che punti a un bucket di output che si trova nella stessa AWS regione del progetto di build.

Errore "Git Clone Failed: unable to access '**your-repository-URL**': SSL certificate problem: self signed certificate" (Git Clone non riuscito: impossibile accedere a "your-repository-URL": problema di certificato SSL: certificato autofirmato)

Problema: quando tenti di eseguire un progetto di compilazione, la compilazione non riesce e restituisce questo errore.

Possibile causa: il repository di origine dispone di un certificato autofirmato, ma non hai scelto l'installazione del certificato dal bucket S3 come parte del progetto di compilazione.

Soluzioni consigliate:

- Modifica il progetto. Per Certificate (Certificato), scegli Install certificate from S3 (Installa certificato da S3). Per Bucket of certificate (Bucket del certificato), scegli il bucket S3 in cui è archiviato il certificato SSL. Per Object key of certificate (Chiave oggetto del certificato), immetti il nome della chiave dell'oggetto S3.
- Modifica il progetto. Seleziona SSL non sicuro per ignorare gli avvisi SSL durante la connessione al repository del progetto Enterprise Server. GitHub

 Note

È consigliabile utilizzare Insecure SSL (SSL non sicuro) solo nella fase di test. Non deve essere utilizzato in un ambiente di produzione.

Errore "The bucket you are attempting to access must be addressed using the specified endpoint" (Il bucket a cui tenti di accedere deve essere individuato tramite l'endpoint specificato) durante l'esecuzione di una compilazione

Problema: quando esegui una compilazione, la fase della compilazione `DOWNLOAD_SOURCE` non riesce con l'errore `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

Possibile causa: il codice sorgente predefinito è archiviato in un bucket S3 e tale bucket si trova in una regione diversa da quella del progetto di compilazione. AWS AWS CodeBuild

Soluzione consigliata: aggiorna le impostazioni del progetto di compilazione in modo che punti a un bucket contenente il codice sorgente precompilato. Assicurati che il bucket si trovi nella stessa AWS regione del progetto di compilazione.

Errore: "This build image requires selecting at least one runtime version." (Questa immagine di compilazione richiede la selezione di almeno una versione runtime.)

Problema: quando esegui una compilazione, la fase della compilazione `DOWNLOAD_SOURCE` non riesce con l'errore `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

Possibile causa: la tua build utilizza la versione 1.0 o successiva dell'immagine standard Amazon Linux 2 (AL2) o la versione 2.0 o successiva dell'immagine standard di Ubuntu e non è specificato un runtime nel file `buildspec`.

Soluzione consigliata: se usi l'immagine `aws/codebuild/standard:2.0` CodeBuild gestita, devi specificare una versione di runtime nella `runtime-versions` sezione del file `buildspec`. Ad esempio, puoi utilizzare il file `buildspec` seguente per un progetto che utilizza PHP:

```
version: 0.2

phases:
  install:
    runtime-versions:
      php: 7.3
  build:
    commands:
      - php --version
artifacts:
  files:
    - README.md
```

Note

Se specifichi una `runtime-versions` sezione e usi un'immagine diversa da Ubuntu Standard Image 2.0 o versione successiva o dall'immagine standard Amazon Linux 2 (AL2) 1.0 o successiva, la build emette l'avviso "Skipping install of runtimes. Runtime version selection is not supported by this build image.»

Per ulteriori informazioni, consulta [Specify runtime versions in the buildspec file.](#)

Errore: "QUEUED: INSUFFICIENT_SUBNET" quando una compilazione ha esito negativo in una coda di compilazione

Problema: una compilazione in una coda di compilazione ha esito negativo con un errore simile a QUEUED: INSUFFICIENT_SUBNET.

Possibili cause: il blocco IPv4 CIDR specificato per il tuo VPC utilizza un indirizzo IP riservato. I primi quattro indirizzi IP e l'ultimo indirizzo IP in ogni blocco CIDR della sottorete non sono disponibili per l'utilizzo e non possono essere assegnati a un'istanza. Ad esempio, in una sottorete con intervallo CIDR 10.0.0.0/24, i cinque indirizzi IP seguenti sono riservati:

- 10.0.0.0:: indirizzo di rete.
- 10.0.0.1: Riservato da AWS per il router VPC.
- 10.0.0.2: Riservato da AWS. L'indirizzo IP del server DNS è sempre la base dell'intervallo di rete VPC più due; tuttavia, riserviamo anche la base di ogni intervallo della sottorete più due. Per VPCs i blocchi CIDR multipli, l'indirizzo IP del server DNS si trova nel CIDR primario. Per ulteriori informazioni, consulta [Amazon DNS Server](#) nella Guida per l'utente di Amazon VPC.
- 10.0.0.3: Riservato da AWS per utilizzi futuri.
- 10.0.0.255: indirizzo di trasmissione di rete. La trasmissione in un VPC non è supportata. Questo indirizzo è riservato.

Soluzioni consigliate: controlla se il VPC utilizza un indirizzo IP riservato. Sostituisci qualsiasi indirizzo IP riservato con uno non riservato. Per ulteriori informazioni, consulta [VPC e dimensionamento della sottorete](#) nella Guida per l'utente di Amazon VPC.

Errore: «Impossibile scaricare la cache: RequestError: Invio della richiesta non riuscito a causa di: x509: impossibile caricare le radici del sistema e nessuna root fornita»

Problema: quando tenti di eseguire un progetto di compilazione, la compilazione non riesce e restituisce questo errore.

Possibile causa: hai configurato il caching come parte del progetto di compilazione e utilizzi un'immagine Docker precedente che include un certificato root scaduto.

Soluzione consigliata: aggiorna l'immagine Docker utilizzata nel progetto. AWS CodeBuild Per ulteriori informazioni, consulta [Immagini Docker fornite da CodeBuild](#).

Errore: «Impossibile scaricare il certificato da S3. AccessDenied»

Problema: quando tenti di eseguire un progetto di compilazione, la compilazione non riesce e restituisce questo errore.

Possibili cause:

- Hai scelto il bucket S3 errato per il certificato.
- Hai immesso la chiave dell'oggetto errata per il certificato.

Soluzioni consigliate:

- Modifica il progetto. Per Bucket of certificate (Bucket del certificato), scegli il bucket S3 in cui è archiviato il certificato SSL.
- Modifica il progetto. Per Object key of certificate (Chiave oggetto del certificato), immetti il nome della chiave dell'oggetto S3.

Errore "Unable to Locate Credentials" (Impossibile individuare le credenziali)

Problema: quando si tenta di eseguire AWS CLI, utilizzare un AWS SDK o chiamare un altro componente simile come parte di una build, si verificano errori di compilazione direttamente correlati all' AWS CLI AWS SDK o al componente. Ad esempio, potresti ricevere un errore di compilazione come `Unable to locate credentials`.

Possibili cause:

- La versione dell' AWS CLI AWS SDK o del componente nell'ambiente di compilazione è incompatibile con. AWS CodeBuild
- Stai eseguendo un contenitore Docker all'interno di un ambiente di compilazione che utilizza Docker e il contenitore non ha accesso alle credenziali per impostazione predefinita. AWS

Soluzioni consigliate:

- Assicurati che il tuo ambiente di compilazione abbia la versione seguente o successiva dell' AWS SDK o del AWS CLI componente.
 - AWS CLI: 1.10.47
 - AWS SDK per C++: 0.2.19
 - AWS SDK for Go: 1.2.5
 - AWS SDK per Java: 1.11.16
 - AWS SDK per: 2.4.7 JavaScript
 - AWS SDK per PHP: 3.18.28
 - AWS SDK per Python (Boto3): 1.4.0
 - AWS SDK per Ruby: 2.3.22
 - Botocore: 1.4.37
 - CoreCLR: 3.2.6-beta
 - Node.js: 2.4.7
- Se devi eseguire un contenitore Docker in un ambiente di compilazione e il contenitore richiede AWS credenziali, devi passare le credenziali dall'ambiente di compilazione al contenitore. Nel file buildspec, includere un comando Docker `run` come il seguente. In questo esempio viene utilizzato il comando `aws s3 ls` per elencare i bucket S3 disponibili. L' `-e` opzione passa attraverso le variabili di ambiente necessarie al contenitore per accedere alle credenziali. AWS

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Se stai creando un'immagine Docker e la build richiede AWS credenziali (ad esempio, per scaricare un file da Amazon S3), devi passare le credenziali dall'ambiente di compilazione al processo di compilazione Docker come segue.
 1. Nel Dockerfile del codice sorgente per l'immagine Docker specificare le seguenti istruzioni ARG.

```
ARG AWS_DEFAULT_REGION
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. Nel file buildspec, includere un comando Docker `build` come il seguente. Le `--build-arg` opzioni impostano le variabili di ambiente necessarie per il processo di compilazione Docker per accedere alle credenziali. AWS

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg  
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -  
t your-image-tag .
```

RequestError errore di timeout durante l'esecuzione CodeBuild in un server proxy

Problema: viene visualizzato un errore RequestError simile a uno dei seguenti:

- RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeoutda CloudWatch Logs.
- Error uploading artifacts: RequestError: send request failed caused by: Put https://*your-bucket*.s3.*your-aws-region*.amazonaws.com/*: dial tcp 52.219.96.208:443: connect: connection refusedda Amazon S3.

Possibili cause:

- ssl-bump non è configurato correttamente.
- La policy di sicurezza della tua organizzazione non consente di utilizzare ssl_bump.
- Il file di specifiche di compilazione non ha impostazioni proxy specificate mediante un elemento proxy.

Soluzioni consigliate:

- Assicurati che ssl-bump sia correttamente configurato. Se per il tuo server proxy utilizzi Squid, consulta [Configura Squid come server proxy esplicito](#).
- Segui questi passaggi per utilizzare endpoint privati per Amazon S3 CloudWatch e Logs:
 1. Nella tabella di routing della sottorete privata, rimuovi la regola aggiunta che instrada sul server proxy il traffico destinato a Internet. Per informazioni, consulta [Creazione di una sottorete nel tuo VPC](#) nella Amazon VPC User Guide.

2. Crea un endpoint Amazon S3 privato e un endpoint CloudWatch Logs e associali alla sottorete privata del tuo Amazon VPC. Per informazioni, consulta i [servizi endpoint VPC](#) nella Amazon VPC User Guide.
 3. Conferma che l'opzione Abilita nome DNS privato nel tuo Amazon VPC è selezionata. Per ulteriori informazioni, consultare [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC.
- Se non utilizzi `ssl-bump` per un server proxy esplicito, aggiungi una configurazione proxy al file di specifiche di compilazione utilizzando un elemento `proxy`. Per ulteriori informazioni, consultare [Esegui CodeBuild in un server proxy esplicito](#) e [Sintassi buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
```

La bourne shell (sh) deve esistere nelle immagini di compilazione

Problema: stai utilizzando un'immagine di build che non è stata fornita da AWS CodeBuild e le tue build hanno esito negativo con il messaggio. `Build container found dead before completing the build`

Possibile causa: La Bourne shell (sh) non è inclusa nell'immagine di build. CodeBuild deve sh eseguire comandi e script di compilazione.

Soluzione consigliata: se sh non è presente nell'immagine di compilazione, assicurati di includerla prima di iniziare altre build che utilizzano l'immagine. (include CodeBuild già le immagini sh di compilazione).

Avvertenza: "Skipping install of runtimes. Runtime version selection is not supported by this build image" (L'installazione dei runtime non viene eseguita. L'immagine di compilazione non supporta la selezione delle versioni dei runtime) durante l'esecuzione di una compilazione

Problema: quando esegui una compilazione, il log di compilazione contiene questa avvertenza.

Possibile causa: la build non utilizza la versione 1.0 o successiva dell'immagine standard Amazon Linux 2 (AL2) o la versione 2.0 o successiva dell'immagine standard di Ubuntu e un runtime è specificato in una `runtime-versions` sezione del file `buildspec`.

Soluzione consigliata: verificare che il file `buildspec` non contenga una sezione `runtime-versions`. La `runtime-versions` sezione è richiesta solo se utilizzi l'immagine standard di Amazon Linux 2 (AL2) o successiva o l'immagine standard di Ubuntu versione 2.0 o successiva.

Errore: «Impossibile verificare JobWorker l'identità» all'apertura della CodeBuild console

Problema: quando si apre la CodeBuild console, viene visualizzato il messaggio di errore «Impossibile verificare l' JobWorker identità».

Possibile causa: il ruolo IAM utilizzato per l'accesso alla console ha un tag `jobId` come chiave. Questa chiave di tag è riservata a CodeBuild e causerà questo errore se presente.

Soluzione consigliata: modifica tutti i tag di ruolo IAM personalizzati che hanno la chiave `jobId` per avere una chiave diversa, ad esempio `jobIdentifier`.

Avvio della compilazione non riuscito

Problema: quando si avvia una build, viene visualizzato un messaggio di errore Build to start.

Possibile causa: è stato raggiunto il numero di build simultanee.

Soluzioni consigliate: attendi il completamento delle altre build oppure aumenta il limite di compilazioni simultanee per il progetto e riavvia la compilazione. Per ulteriori informazioni, consulta [Configurazione del progetto](#).

Accesso ai GitHub metadati nelle build memorizzate nella cache locale

Problema: in alcuni casi, la `directory.git` in una build memorizzata nella cache è un file di testo e non una directory.

Possibili cause: quando la memorizzazione nella cache locale dei sorgenti è abilitata per una build, CodeBuild crea un gitlink per la directory. `.git` Ciò significa che la `.git` directory è in realtà un file di testo contenente il percorso della directory.

Soluzioni consigliate: in tutti i casi, utilizzate il seguente comando per ottenere la directory dei metadati Git. Questo comando funzionerà indipendentemente dal formato di `.git`:

```
git rev-parse --git-dir
```

AccessDenied: Il proprietario del bucket per il gruppo di report non corrisponde al proprietario del bucket S3...

Problema: quando carica i dati di test su un bucket Amazon S3 CodeBuild , non è in grado di scrivere i dati di test nel bucket.

Possibili cause:

- L'account specificato per il proprietario del bucket del gruppo di report non corrisponde al proprietario del bucket Amazon S3.
- Il ruolo di servizio non dispone dell'accesso in scrittura al bucket.

Soluzioni consigliate:

- Modifica il proprietario del bucket del gruppo di report in modo che corrisponda al proprietario del bucket Amazon S3.
- Modifica il ruolo del servizio per consentire l'accesso in scrittura al bucket Amazon S3.

Errore: «Le tue credenziali non dispongono di uno o più ambiti di privilegi richiesti» durante la creazione di un progetto con CodeBuild CodeConnections

Problema: quando crei un CodeBuild progetto con CodeConnections, non sei autorizzato a installare un webhook Bitbucket.

Possibili cause:

- Il nuovo ambito di autorizzazione potrebbe non essere stato accettato nel tuo account Bitbucket.

Soluzioni consigliate:

- Per accettare la nuova autorizzazione, dovresti aver ricevuto un'e-mail con oggetto intitolato Action required - Scopes for AWS CodeStar have change sent by Bitbucket,. notifications-noreply@bitbucket.org L'e-mail contiene un link per concedere le autorizzazioni del webhook all'installazione esistente dell'app Bitbucket. CodeConnections
- Se non riesci a trovare l'email, puoi concedere l'autorizzazione accedendo a https://bitbucket.org/site/addons/reauthorize?addon_key=aws-codestar selezionando l'area di lavoro a https://bitbucket.org/site/addons/reauthorize?account=<workspace-name>&addon_key=aws-codestar cui desideri concedere l'autorizzazione al webhook.

**AWS CodeStar requests access**

This app is hosted at <https://codestar-connections.webhooks.aws>

Read your account information

Read and modify your repositories and their pull requests

Administer your repositories

Read and modify your repositories' webhooks

Authorize for workspace

Allow AWS CodeStar to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

Grant access Cancel

Errore: «Siamo spiacenti, non è stato richiesto alcun terminale, impossibile ricevere input» durante la compilazione con il comando di installazione di Ubuntu

Problema: [se esegui build privilegiate di contenitori GPU, è possibile che tu stia installando NVIDIA Container Toolkit seguendo questa procedura](#). CodeBuild Nell'ultima versione dell'immagine, CodeBuild preinstalla e configura docker nvidia-container-toolkit con l'immagine più recente e curata amazonlinux.ubuntu Seguendo questa procedura, le build con il comando di installazione di Ubuntu falliranno con il seguente errore:

```
Running command curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | gpg --
dearmor --no-tty -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
gpg: Sorry, no terminal at all requested - can't get input
curl: (23) Failed writing body
```

Possibili cause: la chiave gpg esiste già nella stessa posizione.

Soluzioni consigliate: `nvidia-container-toolkit` è già installato nell'immagine. Se vedi questo errore, puoi saltare il processo di installazione e riavvio del docker nel tuo buildspec.

Quote per AWS CodeBuild

Le tabelle seguenti elencano le quote correnti in AWS CodeBuild. Queste quote si riferiscono a ciascuna AWS regione supportata per ogni AWS account, se non diversamente specificato.

Quote di servizio

Di seguito sono riportate le quote predefinite per il AWS CodeBuild servizio.

Name	Predefinita	Adattata	Description
Tag associati per progetto	Ogni Regione supportata: 50	No	Numero massimo di tag associabili a un progetto di compilazione
Costruisci progetti	Ogni regione supportata: 5.000	Sì	Numero massimo di progetti di compilazione
Timeout build in minuti	Ogni regione supportata: 2.160	No	Timeout massimo di build in minuti
Richiesta simultanea di informazioni sulle build	Ogni regione supportata: 100	No	Numero massimo di build su cui puoi richiedere informazioni in qualsiasi momento utilizzando la AWS CLI o AWS un SDK.
Richieste simultanee di informazioni sui progetti di build	Ogni regione supportata: 100	No	Numero massimo di progetti di compilazione su cui puoi richiedere informazioni in qualsiasi momento utilizzando la AWS CLI o un AWS SDK.
Esecuzione simultanea di build per ambienti ARM Lambda/10 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultane

Name	Predefinita	Adattata	Description
			a per l'ambiente ARM Lambda/10 GB
Esecuzione simultanea di build per ambienti ARM Lambda/1 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM Lambda/1 GB
Esecuzione simultanea di build per ambienti ARM Lambda/2 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM Lambda/2 GB
Esecuzione simultanea di build per ambienti ARM Lambda/4 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM Lambda/4 GB
Esecuzione simultanea di build per ambienti ARM Lambda/8 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM Lambda/8 GB
Compilazioni in esecuzione simultanea per l'ambiente ARM/2 XLarge	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM/2 XLarge
Compilazioni in esecuzione simultanea per l'ambiente ARM/Large	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM/Large

Name	Predefinita	Adattate	Description
Compilazioni per ambiente in esecuzione e simultanea ARM/Medium	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM/Medium
Compilazioni per ambiente in esecuzione e simultanea ARM/Small	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM/Small
Compilazioni per ambiente in esecuzione e simultanea ARM/XLarge	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente ARM/XLarge
Compilazioni in esecuzione simultanea per GPU Linux Ampio ambiente	Ogni regione supportata: 0	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux GPU/Large
Compilazioni in esecuzione simultanea per ambienti GPU Linux Small	Ogni regione supportata: 0	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux GPU/Small
Esecuzione simultanea di build per ambienti Linux Lambda/10 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux Lambda/10 GB
Esecuzione simultanea di build per ambienti Linux Lambda/1 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux Lambda/1 GB

Name	Predefinita	Adattata	Description
Esecuzione simultanea di build per ambienti Linux Lambda/2 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux Lambda/2 GB
Esecuzione simultanea di build per ambienti Linux Lambda/4 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux Lambda/4 GB
Esecuzione simultanea di build per ambienti Linux Lambda/8 GB	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux Lambda/8 GB
Compilazioni in esecuzione simultanea per l'ambiente Linux/2 XLarge	Ogni regione supportata: 0	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux/2 XLarge
Compilazioni per ambiente in Linux/Large esecuzione simultanea	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux/Large
Compilazioni per ambiente in esecuzione simultanea Linux/Medium	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux/Medium
Compilazioni per ambiente in esecuzione simultanea Linux/Small	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux/Small

Name	Predefinita	Adattate	Description
Compilazioni per ambiente in esecuzione e simultanea Linux/XLarge	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Linux/XLarge
Build in esecuzione simultanea per Windows Server 2019/Ambiente di grandi dimensioni	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per ambiente Windows Server 2019/Large
Build in esecuzione simultanea per ambiente Windows Server 2019/Medium	Ogni regione supportata: 1	Sì	Numero massimo di build eseguite contemporaneamente per l'ambiente Windows Server 2019/Medium
Build in esecuzione simultanea per l'ambiente Windows Server 2022/2 XLarge	Ogni regione supportata: 1	Sì	Numero massimo di build eseguite contemporaneamente per l'ambiente Windows Server 2022/2 XLarge
Build in esecuzione simultanea per Windows Server 2022/Ambiente di grandi dimensioni	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per Windows Server 2022/Ambiente di grandi dimensioni
Build in esecuzione simultanea per ambiente Windows Server 2022/Medium	Ogni regione supportata: 1	Sì	Numero massimo di build eseguite contemporaneamente per l'ambiente Windows Server 2022/Medium

Name	Predefinita	Adattata	Description
Build in esecuzione simultanea per l'ambiente Windows Server 2022/ XLarge	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Windows Server 2022/ XLarge
Compilazioni in esecuzione simultanea per l'ambiente Windows/Large	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Windows/Large
Compilazioni per ambiente in esecuzione simultanea Windows/Medium	Ogni regione supportata: 1	Sì	Numero massimo di build in esecuzione simultanea per l'ambiente Windows/Medium
Periodo minimo per il timeout di build in minuti	Ogni Regione supportata: 5	No	Timeout minimo di build in minuti
Gruppi di sicurezza nella configurazione VPC	Ogni Regione supportata: 5	No	Gruppi di sicurezza disponibili per configurazione VPC
Sottoreti nella configurazione VPC	Ogni regione supportata: 16	No	Sottoreti disponibili per configurazione VPC

Note

Le metriche interne determineranno le quote predefinite per le build in esecuzione simultanea.

Le quote per il numero massimo di build in esecuzione simultanea variano a seconda del tipo di elaborazione. Per alcune piattaforme e alcuni tipi di calcolo, l'impostazione predefinita è 20. Per richiedere una quota di build simultanee più elevata o se ricevi il messaggio di errore «Impossibile

avere più di X build attive per l'account», usa il link qui sopra per effettuare la richiesta. [Per ulteriori informazioni sui prezzi, consulta AWS CodeBuild la pagina dei prezzi.](#)

Altri limiti

Costruisci progetti

Risorsa	Di default
Caratteri consentiti nella descrizione di un progetto di compilazione	Qualsiasi
Caratteri consentiti nel nome di un progetto di compilazione	Le lettere A-Z e a-z, i numeri 0-9 e i caratteri speciali - e _
Lunghezza del nome di un progetto di compilazione	Da 2 a 150 caratteri, inclusi
Lunghezza massima della descrizione di un progetto di compilazione	255 caratteri
Numero massimo di report che è possibile aggiungere a un progetto	5
Numero di minuti specificabili in un progetto di compilazione per il timeout di tutte le compilazioni correlate	Da 5 a 2160 (36 ore)

Costruisce

Risorsa	Di default
Tempo massimo di conservazione della cronologia di una compilazione	1 anno

Risorsa	Di default
Numero di minuti specificabili in un progetto di compilazione per il timeout di una singola compilazione	Da 5 a 2160 (36 ore)

Flotte di calcolo

Risorsa	Di default
Numero simultaneo di flotte di elaborazione	10
Istanze in esecuzione simultanea per flotte ambientali ARM/Small	1
Istanze in esecuzione simultanea per flotte ambientali ARM/Large	1
Istanze in esecuzione simultanea per flotte ambientali Linux/Small	1
Istanze in esecuzione simultanea per flotte ambientali Linux/Medium	1
Istanze in esecuzione simultanea per flotte ambientali Linux/Large	1
Istanze in esecuzione simultanea per flotte ambientali Linux/XLarge	1
Istanze in esecuzione simultanea per flotte di ambienti Linux/2 XLarge	0
Istanze in esecuzione simultanea per flotte di ambienti Linux GPU/Small	0
Istanze in esecuzione simultanea per flotte di ambienti Linux GPU/Large	0

Risorsa	Di default
Istanze in esecuzione simultanea per flotte di ambienti Windows Server 2019/Medium	1
Istanze in esecuzione simultanea per Windows Server 2019/Flotte di ambienti di grandi dimensioni	1
Istanze in esecuzione simultanea per flotte di sistemi Windows Server 2022/Ambiente medio	1
Istanze in esecuzione simultanea per Windows Server 2022/Flotte con ambienti di grandi dimensioni	1
Istanze in esecuzione simultanea per flotte di ambienti Mac ARM/Medium	1
Istanze in esecuzione simultanea per flotte di ambienti Mac ARM/Large	1

Report

Risorsa	Di default
Durata massima della disponibilità di un report di test dopo la creazione	30 giorni
Lunghezza massima di un messaggio di test case	5.000 caratteri
Lunghezza massima del nome di un test case	1.000 caratteri
Numero massimo di gruppi di report per AWS account	5.000
Numero massimo di casi di test per report	500

Tag

I limiti di tag si applicano ai tag sui progetti di CodeBuild CodeBuild compilazione e sulle risorse dei gruppi di report.

Risorsa	Di default
Nomi chiave tag risorsa	Qualsiasi combinazione di lettere Unicode, numeri, spazi e caratteri consentiti in UTF-8 con una lunghezza compresa tra 1 e 127 caratteri. I caratteri consentiti sono + - = . _ : / @ I nomi delle chiavi di tag devono essere univoci e ogni chiave può avere un solo valore. Il nome di una chiave tag non può: - iniziare con aws : - contenere solo spazi - terminare con uno spazio - contenere emojis o uno dei seguenti caratteri : ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Valori del tag della risorsa	Qualsiasi combinazione di lettere Unicode, numeri, spazi e caratteri consentiti in UTF-8 con una lunghezza compresa tra 0 e 255 caratteri. I caratteri consentiti sono + - = . _ : / @ Una chiave può avere un solo valore, ma molte chiavi possono avere lo stesso valore. Un valore di chiave tag non può contenere emoji o uno dei seguenti caratteri: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;

AWS CodeBuild Cronologia dei documenti della Guida dell'utente

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di AWS CodeBuild. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile iscriversi a un feed RSS.

- Ultima versione dell'API: 2016-10-06

Modifica	Descrizione	Data
Contenuto aggiornato: politiche AWS gestite (predefinite) per AWS CodeBuild	Le AWSCode BuildDeveloperAccess politiche AWSCode BuildAdminAccess e sono state aggiornate per includere l'AmazonOpenDataChannel autorizzazione richiesta per il debug interattivo della build di Session Manager.	1 dicembre 2025
Pull, richiedi l'approvazione del commento	CodeBuild supporta le politiche di compilazione delle pull request che forniscono un controllo aggiuntivo sulle build attivate dalle richieste pull.	7 agosto 2025
Contenuto aggiornato: politiche AWS gestite (predefinite) per AWS CodeBuild	Le AWSCode BuildReadOnlyAccess politiche AWSCode BuildAdminAccess AWSCodeBuildDeveloperAccess, e sono state aggiornate. La risorsa originale <code>arn:aws:codeconnections:*:*:connection/*</code> è stata aggiornata	10 giugno 2025

`aarn:aws:codeconnec
tions:*:*:*` .

[Nuovo riferimento per le chiavi di CodeBuild condizione](#)

È stata aggiunta una nuova pagina di riferimento con esempi per l'utilizzo CodeBuild delle chiavi di condizione. Vedi le [chiavi di AWS CodeBuild condizione](#).

15 maggio 2025

[Nuovi contenuti: Esempio di server di creazione di immagini Docker per CodeBuild](#)

CodeBuild ora supporta l'offload delle build Docker su un server di creazione di immagini gestito.

15 maggio 2025

[Nuovo tipo di calcolo: CUSTOM_INSTANCE_TYPE](#)

CodeBuild ora consente di creare una flotta di capacità riservata con un tipo di istanza specifico utilizzando CUSTOM_INSTANCE_TYPE .

23 aprile 2025

[Nuovo supporto per CodeBuild sandbox](#)

Sono state aggiunte informazioni sull'utilizzo della nuova CodeBuild sandbox. Vedi [Debug build with sandbox. CodeBuild](#)

7 aprile 2025

[Nuovi tipi di ambiente Windows](#)

CodeBuild ora supporta i tipi di ambiente Windows XL e 2XL. Per ulteriori informazioni, consulta [Tipi di calcolo di ambiente di compilazione](#).

31 marzo 2025

Caching Amazon S3 aggiornato	CodeBuild ora supporta un nuovo comportamento di caching per la memorizzazione nella cache di Amazon S3.	28 marzo 2025
Nuovo contenuto: opzioni di configurazione di GitHub Actions Runner	CodeBuild ora supporta CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME la registrazione a livello aziendale.	11 marzo 2025
Nuovo contenuto: aggiungi un nuovo tipo di filtro webhook	Aggiunge il supporto per un nuovo tipo di filtro webhook (<code>()ORGANIZATION_NAME</code>).	11 marzo 2025
Nuovi contenuti: tutorial per la firma del codice Apple con Fastlane con archiviazione dei certificati S3	Aggiungi un nuovo tutorial per la firma del codice Apple con Fastlane sull'uso di CodeBuild di S3 per l'archiviazione dei certificati	5 febbraio 2025
Nuovi contenuti: tutorial per la firma del codice Apple con Fastlane con GitHub archiviazione dei certificati	Aggiungi un nuovo tutorial per l'uso di CodeBuild della firma del codice Apple con Fastlane GitHub per l'archiviazione dei certificati	5 febbraio 2025
Nuovo contenuto: Buildkite runner	Aggiungi nuovi contenuti per il runner Buildkite	31 gennaio 2025
Nuovo contenuto: webhook manuali di Buildkite	Aggiungi il supporto per i webhook manuali di Buildkite.	31 gennaio 2025

Nuovo contenuto: Batch build buildspec reference	Aggiungi il supporto per le build in batch in flotte a capacità riservata e ambienti Lambda.	08 gennaio 2025
Nuovi contenuti: esegui test paralleli nelle build in batch	Aggiungi nuovi contenuti per i test paralleli nelle build in batch.	2 gennaio 2025
Nuovi contenuti: riprova le build automaticamente	CodeBuild ora supporta il ritentativo automatico per le build di webhook.	18 dicembre 2024
Nuovi contenuti: configura una credenziale di registro privata per i corridori ospitati autonomamente	Aggiunge il supporto per l'impostazione delle credenziali di registro quando si utilizzano immagini personalizzate da registri non privati.	13 dicembre 2024
Nuovo contenuto: GitHub opzioni di configurazione di Actions Runner	CodeBuild GitHub I corridori ospitati autonomamente da Actions ora ti consentono di registrare i tuoi corridori a livello di organizzazione e configurare un ID di gruppo di corridori specifico.	12 dicembre 2024
Nuovo contenuto: aggiungi l'attributo on-failure RETRY	CodeBuild ora ti consente di configurare un attributo on-failure RETRY nel tuo buildspec.	12 dicembre 2024
Nuovi contenuti: webhook manuali GitLab	Aggiungi il supporto per i webhook GitLab manuali.	11 dicembre 2024
Contenuto aggiornato: alias aggiornati	Aggiorna gli alias per le immagini di runtime standard basate su Linux.	22 novembre 2024

Contenuto aggiornato: le sostituzioni delle etichette sono supportate con -hosted runner CodeBuild GitLab	Aggiunge il supporto per le sostituzioni personalizzate delle etichette con immagini per i corridori. GitLab	22 novembre 2024
Contenuto aggiornato: le sostituzioni delle etichette sono supportate con il runner -hosted Actions CodeBuild GitHub	Aggiunge il supporto per le sostituzioni personalizzate delle etichette di immagine per i runner Actions. GitHub	22 novembre 2024
Contenuto aggiornato: politiche AWS gestite (predefinite) per AWS CodeBuild	Le AWSCode BuildRead OnlyAccess politiche AWSCode BuildAdminAccess AWSCodeBuildDeveloperAccess, e sono state aggiornate. La risorsa originale <code>arn:aws:codebuild:*</code> è stata aggiornata a <code>arn:aws:codebuild:*:*:project/*</code> .	15 novembre 2024
Contenuto aggiornato: Capacità riservata	Le flotte con capacità riservata ora supportano build non containerizzate: ARM EC2, Linux EC2 e Windows EC2.	12 novembre 2024
Contenuto aggiornato: capacità riservata	Le flotte con capacità riservata ora supportano l'elaborazione basata sugli attributi.	6 novembre 2024
Nuovi contenuti: riprova le build automaticamente	CodeBuild ora ti consente di abilitare i tentativi automatici per le tue build.	25 ottobre 2024

Nuovi contenuti: esegui CodeBuild in un server proxy gestito per flotte a capacità riservata	Aggiungi il supporto per le configurazioni proxy per flotte a capacità riservata.	15 ottobre 2024
Nuovi contenuti: corridori autogestiti GitLab	Aggiungi nuovi contenuti per i corridori autogestiti GitLab	17 settembre 2024
Nuovi contenuti: GitLab webhook di gruppo	Aggiungi il supporto per i webhook di GitLab gruppo.	17 settembre 2024
Nuovi contenuti: esegui i comandi buildspec nelle fasi INSTALL, PRE_BUILD e POST_BUILD	-with-buildspec Aggiungi il supporto per.	20 agosto 2024
Contenuto aggiornato: capacità riservata	Le flotte a capacità riservata ora supportano macOS.	19 agosto 2024
Nuovi contenuti: GitHub connessioni alle app	Aggiungi il supporto per le connessioni alle GitHub app.	14 agosto 2024
Nuovi contenuti: connessioni all'app Bitbucket	Aggiungi il supporto per le connessioni all'app Bitbucket.	14 agosto 2024
Nuovi contenuti: Token di accesso multiplo in CodeBuild	Aggiungi il supporto per l'approvvigionamento di token di accesso a fornitori di terze parti da segreti all'interno Gestione dei segreti AWS o tramite AWS CodeConnections connessioni.	14 agosto 2024
Contenuto aggiornato: capacità riservata	Le flotte con capacità riservata ora supportano i tipi di XLarge elaborazione ARM Medium XLarge, ARM e ARM 2.	5 agosto 2024

Contenuto aggiornato: capacità riservata	CodeBuild ora supporta la connettività VPC per flotte a capacità riservata su Windows.	1° agosto 2024
Nuovi tipi di elaborazione ARM	CodeBuild ora supporta i tipi di XLarge calcolo ARM Medium XLarge, ARM e ARM 2. Per ulteriori informazioni, consulta Tipi di calcolo di ambiente di compilazione .	10 luglio 2024
Contenuto aggiornato: firma SHA	Aggiorna la firma Secure Hash Algorithm (SHA) per x86_64 e ARM.	19 giugno 2024
Nuovi contenuti: webhook globali e organizzativi GitHub	Aggiungi il supporto per i GitHub webhook globali e organizzativi.	17 giugno 2024
Nuovi contenuti: aggiungi un nuovo tipo di filtro webhook	Aggiunge il supporto per un nuovo tipo di filtro webhook <code>()REPOSITORY_NAME</code> .	17 giugno 2024
Spazio su disco aggiornato	I tipi di ARM Large calcolo ARM Small e ora dispongono di uno spazio su disco maggiore.	4 giugno 2024
Nuovi contenuti: webhook GitHub manuali	Aggiungi il supporto per i webhook GitHub manuali.	23 maggio 2024
Contenuto aggiornato: capacità riservata	CodeBuild ora supporta la connettività VPC per flotte a capacità riservata su Amazon Linux.	15 maggio 2024

Contenuto aggiornato: Immagini di calcolo Lambda	Aggiungere il supporto Lambda per .NET 8 (a1-lambda/aarch64/dotnet8 e) a1-lambda/x86_64/dotnet8	8 maggio 2024
Quota aggiornata: timeout di compilazione	Aggiorna la quota massima di timeout di compilazione a 2160 minuti (36 ore).	1° maggio 2024
Contenuto aggiornato: politiche AWS gestite (predefinite) per AWS CodeBuild	Le AWSCodeBuildReadOnlyAccess politiche OnlyAccess politiche AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess, e sono state aggiornate per riflettere il AWS CodeConnections rebranding.	30 aprile 2024
Nuovi contenuti: Password o token di accesso dell'app Bitbucket	Aggiungi il supporto per i token di accesso Bitbucket.	11 aprile 2024
Nuovi contenuti: Individuazione automatica dei report in CodeBuild	CodeBuild ora supporta l'individuazione automatica dei report.	4 aprile 2024
Nuovi contenuti: Actions runner ospitati autonomamente GitHub	Aggiungi nuovi contenuti per gli Actions runner ospitati autonomamente GitHub	2 aprile 2024
Nuovi contenuti: connessioni GitLab	Aggiungi supporto GitLab e connessioni GitHub autogestite.	25 marzo 2024

Nuovi contenuti: aggiungi nuovi eventi webhook e tipi di filtri	Aggiunge il supporto per nuovi eventi webhook (RELEASEDandPRERELEASED) e tipi di filtro (TAG_NAMEandRELEASE_NAME).	15 marzo 2024
Nuovo contenuto: aggiungi un nuovo evento webhook: PULL_REQUEST_CLOSED	Aggiungi il supporto per un nuovo evento webhook: PULL_REQUEST_CLOSED	20 febbraio 2024
Contenuto aggiornato: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per Windows Server Core 2019 (windows-base:2019-3.0)	7 febbraio 2024
Contenuto aggiornato: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per i nuovi runtime per Amazon Linux 2023 () a12/aarch64/standard/3.0	29 gennaio 2024
Nuovo contenuto: capacità riservata	CodeBuild ora supporta flotte a capacità riservata in CodeBuild.	18 gennaio 2024
Nuovo tipo di elaborazione	CodeBuild ora supporta un tipo di XLarge calcolo Linux. Per ulteriori informazioni, consulta Tipi di calcolo di ambiente di compilazione .	8 gennaio 2024
Contenuto aggiornato: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per nuovi runtime per Amazon Linux 2 (a12/standard/5.0) e Ubuntu (ubuntu/standard/7.0)	14 dicembre 2023

Contenuti aggiornati: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per nuove immagini di calcolo Lambda	8 dicembre 2023
Nuovi contenuti: calcolo AWS Lambda	Aggiungi nuovi contenuti per il calcolo AWS Lambda	6 novembre 2023
Contenuto aggiornato: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per Amazon Linux 2 (a12/standard/5.0)	17 maggio 2023
Modifiche alle politiche gestite per CodeBuild	I dettagli sugli aggiornamenti alle politiche AWS gestite per CodeBuild sono ora disponibili. Per ulteriori informazioni, consulta l'argomento relativo agli Aggiornamenti di CodeBuild sulle policy gestite da AWS .	16 maggio 2023
Contenuti aggiornati: Immagini Docker fornite da CodeBuild	Rimuovi il supporto per Amazon Linux 2 (a12/standard/3.0) e aggiungi il supporto per Amazon Linux 2 (a12/standard/corretto8) e Amazon Linux 2 (a12/standard/corretto11)	9 maggio 2023
Contenuti aggiornati: Immagini Docker fornite da CodeBuild	Aggiungi il supporto per Ubuntu 22.04 (ubuntu/standard/7.0	13 aprile 2023
Contenuto aggiornato: Immagini Docker fornite da CodeBuild	Rimuovi il supporto per Ubuntu 18.04 (ubuntu/standard/4.0) e Amazon Linux 2 (a12/aarch64/standard/1.0)	31 marzo 2023

[Contenuto aggiornato: rimuovi la limitazione VPC](#)

Rimozione della seguente limitazione: se si configura CodeBuild l'utilizzo di un VPC, la memorizzazione nella cache locale non è supportata. A partire dal 28/02/22, la build del VPC richiederà più tempo poiché verrà utilizzata una nuova istanza Amazon EC2 per ogni build.

1 marzo 2023

[Contenuti aggiornati: Immagini Docker fornite da CodeBuild](#)

Rimuovi il supporto per Ubuntu 18.04 (ubuntu/standard/3.0) e Amazon Linux 2 (al2/standard/2.0)

30 giugno 2022

[Esempio di Amazon ECR: limitazione dell'accesso alle immagini](#)

Quando vengono utilizzate CodeBuild credenziali per estrarre un'immagine Amazon ECR, puoi limitare l'accesso all'immagine a un progetto specifico CodeBuild . Per ulteriori informazioni, consulta [l'esempio di Amazon ECR](#).

10 marzo 2022

[Aggiunta del supporto regionale](#)

Il tipo di ARM_CONTAINER elaborazione è ora supportato nelle seguenti regioni aggiuntive: Asia Pacifico (Seoul), Canada (Centrale), Europa (Londra) ed Europa (Parigi). Per ulteriori informazioni, consulta [Tipi di calcolo di ambiente di compilazione](#).

10 marzo 2022

Nuova limitazione VPC	Se si configura CodeBuild l'utilizzo di un VPC, la memorizzazione nella cache locale non è supportata. A partire dal 28/02/22, la build del VPC richiederà più tempo poiché verrà utilizzata una nuova istanza Amazon EC2 per ogni build.	25 febbraio 2022
Modalità report Batch	CodeBuild ora consente di selezionare in che modo gli stati di compilazione in batch vengono inviati al fornitore di origine di un progetto. Per ulteriori informazioni, consulta Modalità report Batch .	4 ottobre 2021
Nuovo tipo di calcolo	CodeBuild ora supporta un tipo di calcolo ARM di piccole dimensioni. Per ulteriori informazioni, consulta Tipi di calcolo di ambiente di compilazione .	13 settembre 2021
progetti di costruzione pubblica	CodeBuild ora ti consente di rendere disponibili al pubblico i risultati di compilazione dei tuoi progetti di compilazione senza richiedere l'accesso a un AWS account. Per ulteriori informazioni, consulta Public build projects .	11 agosto 2021

[Debug delle sessioni per le build in batch](#)

CodeBuild ora supporta il debug delle sessioni per le build in batch. [Per ulteriori informazioni, consulta build-graph e build-list.](#)

3 marzo 2021

[Limite di compilazione simultanea a livello di progetto](#)

CodeBuild ora consente di limitare il numero di build simultanee per un progetto di compilazione. Per ulteriori informazioni, consulta [Configurazione del progetto](#) e [concurrentBuildLimit](#)

16 febbraio 2021

[Nuova proprietà buildspec: s3-prefix](#)

CodeBuild ora fornisce la proprietà `s3-prefix` buildspec per gli artefatti che consente di specificare un prefisso di percorso per gli artefatti che vengono caricati su Amazon S3. [Per](#) ulteriori informazioni, consulta `s3-prefix`.

09 febbraio 2021

[Nuova proprietà buildspec: on-failure](#)

CodeBuild ora fornisce la proprietà `on-failure` buildspec per le fasi di compilazione che consente di determinare cosa succede quando una fase di compilazione fallisce. [Per ulteriori informazioni, vedere on-failure.](#)

09 febbraio 2021

Nuova proprietà buildspec: exclude-paths	CodeBuild ora fornisce la proprietà <code>exclude-paths</code> buildspec per gli artefatti che consente di escludere i percorsi dagli artefatti di costruzione. Per ulteriori informazioni, consulta <code>exclude-paths</code>.	09 febbraio 2021
Nuova proprietà buildspec: enable-symlinks	CodeBuild ora fornisce la proprietà <code>enable-symlinks</code> buildspec per gli artefatti che consente di conservare i collegamenti simbolici in un elemento ZIP. Per ulteriori informazioni, vedere <code>enable-symlinks</code> .	09 febbraio 2021
Miglioramento del nome degli artefatti Buildspec	CodeBuild ora consente alla proprietà di contenere informazioni sul percorso. <code>artifacts/name</code> Per ulteriori informazioni, vedere name .	09 febbraio 2021
Reportistica sulla copertura del codice	CodeBuild ora fornisce report sulla copertura del codice. Per ulteriori informazioni, consulta i report sulla copertura del codice .	30 luglio 2020
Compilazioni in Batch	CodeBuild ora supporta l'esecuzione di build simultane e coordinate di un progetto. Per ulteriori informazioni, consulta Batch builds in CodeBuild .	30 luglio 2020

[Immagine Windows Server 2019](#)

CodeBuild ora fornisce un'immagine di build di Windows Server Core 2019. Per ulteriori informazioni, consulta le [immagini Docker fornite da CodeBuild](#).

20 luglio 2020

[Gestore di sessioni](#)

CodeBuild ora consente di mettere in pausa una build in esecuzione e quindi utilizzare AWS Systems Manager Session Manager per connettersi al contenitore di compilazione e visualizzare lo stato del contenitore. Per ulteriori informazioni, consulta [Session Manager](#).

20 luglio 2020

[Argomento aggiornato](#)

CodeBuild ora supporta la specificazione di una shell da utilizzare nei propri ambienti di compilazione nel file buildspec. [Per ulteriori informazioni, consulta il riferimento alle specifiche di Build](#).

25 giugno 2020

[Reportistica dei test con framework di test](#)

Sono stati aggiunti diversi argomenti che descrivono come generare report di CodeBuild test con diversi framework di test. Per ulteriori informazioni, vedere [Report di test con framework di test](#).

29 maggio 2020

Argomenti aggiornati	CodeBuild ora supporta l'aggiunta di tag ai gruppi di report. Per ulteriori informazioni, consulta ReportGroup .	21 maggio 2020
Support per la reportistica dei test	CodeBuild il supporto per la rendicontazione dei test è ora disponibile a livello generale.	21 maggio 2020
Argomenti aggiornati	CodeBuild ora supporta la creazione di filtri create webhook per Github e Bitbucket che attivano le build solo quando il messaggio di head commit corrisponde all'espressione specificata. Per maggiori informazioni, consulta l'esempio di GitHub pull request e di filtro webhook e l'esempio di pull request e webhook di Bitbucket pull request e webhook.	6 maggio 2020
Nuovi argomenti	CodeBuild ora supporta la condivisione di risorse per progetti di compilazione e gruppi di report. Per ulteriori informazioni, consulta Utilizzo dei progetti condivisi e Utilizzo dei gruppi di progetti condivisi .	13 dicembre 2019

Argomenti nuovi e aggiornati

CodeBuild ora supporta il reporting dei test durante l'esecuzione di un progetto di compilazione. Per ulteriori informazioni, consulta [Lavorare con i report sui test](#), [Creare un rapporto di test](#) e [Creare un rapporto di test utilizzando l' AWS CLI esempio](#).

25 novembre 2019

Argomento aggiornato

CodeBuild ora supporta i tipi di ambiente Linux GPU e Arm e il tipo di 2xlarge elaborazione. Per ulteriori informazioni, consulta [Tipi di calcolo di ambiente di compilazione](#).

19 novembre 2019

Argomenti aggiornati

CodeBuild ora supporta i numeri di build su tutte le build, l'esportazione di variabili di ambiente e Gestione dei segreti AWS l'integrazione. Per ulteriori informazioni, consulta [Variabili esportate](#) e [Secrets Manager](#) nella [Sintassi delle specifiche di compilazione](#).

6 novembre 2019

Nuovo argomento	CodeBuild ora supporta le regole di notifica. È possibile utilizzare le regole di notifica per notificare agli utenti modifiche importanti nei progetti di creazione. Per ulteriori informazioni, consulta Creazione di una regola di notifica .	05 novembre 2019
Argomenti aggiornati	CodeBuild ora supporta i runtime Android versione 29 e Go versione 1.13. Per ulteriori informazioni, consulta Immagini Docker fornite da CodeBuild e Sintassi buildspec .	10 settembre 2019
Argomenti aggiornati	Quando crei un progetto, ora puoi scegliere l'immagine gestita di Amazon Linux 2 (AL2). Per ulteriori informazioni, consulta Immagini Docker fornite da CodeBuild e Versioni di runtime nell'esempio di file buildspec per CodeBuild .	16 agosto 2019
Argomento aggiornato	Ora, quando crei un progetto, puoi scegliere di disabilitare la crittografia dei log S3 e, se utilizzi un repository sorgente basato su Git, puoi includere moduli secondari Git. Per ulteriori informazioni, consulta Creare un progetto di compilazione in CodeBuild .	8 marzo 2019

Nuovo argomento	CodeBuild ora supporta la memorizzazione nella cache locale. Al momento della creazione di una compilazione è possibile specificare il caching locale basato su una o più delle quattro modalità disponibili. Per ulteriori informazioni, consulta Build caching in. CodeBuild	21 febbraio 2019
Nuovi argomenti	CodeBuild ora supporta i gruppi di filtri webhook per specificare gli eventi che attivano una build. Per ulteriori informazioni, consulta Filtrare gli eventi GitHub webhook e Filtrare gli eventi webhook di Bitbucket .	8 febbraio 2019
Nuovo argomento	La Guida CodeBuild per l'utente ora mostra come utilizzarla CodeBuild con un server proxy. Per ulteriori informazioni, vedere Utilizzo CodeBuild con un server proxy .	4 febbraio 2019

Argomenti aggiornati	CodeBuild ora supporta l'utilizzo di un'immagine Amazon ECR che si trova in un altro AWS account. Diversi argomenti sono stati aggiornati per riflettere questa modifica, tra cui Amazon ECR sample for CodeBuild , Create a build project e Create a CodeBuild service role .	24 gennaio 2019
Support per registri Docker privati	CodeBuild ora supporta l'utilizzo di un'immagine Docker archiviata in un registro privato come ambiente di runtime. Per ulteriori informazioni, consulta Registro privato con Gestione dei segreti AWS esempio .	24 gennaio 2019
Argomento aggiornato	CodeBuild ora supporta l'utilizzo di un token di accesso per connettersi ai GitHub repository (con un token di accesso personale) e Bitbucket (con una password per l'app). Per ulteriori informazioni, vedi Creazione di un progetto di compilazione (console) e Utilizzo dei token di accesso con il tuo provider di origine .	6 dicembre 2018
Argomento aggiornato	CodeBuild ora supporta nuove metriche di build che misurano la durata di ogni fase di una build. Per ulteriori informazioni, consulta le CodeBuild CloudWatchmetriche .	15 novembre 2018

Argomento relativo alla policy degli endpoint VPC	Gli endpoint Amazon VPC per CodeBuild ora supportano le politiche. Per ulteriori informazioni, consulta Creazione di una policy di endpoint VPC per CodeBuild .	9 novembre 2018
Contenuti aggiornati	Gli argomenti sono stati aggiornati in base alla nuova console.	30 ottobre 2018
Esempio di Amazon EFS	CodeBuild può montare un file system Amazon EFS durante una compilazione utilizzando i comandi nel file buildspec di un progetto. Per ulteriori informazioni, consulta l'esempio di Amazon EFS per CodeBuild .	26 ottobre 2018
Webhook Bitbucket	CodeBuild ora supporta i webhook quando usi Bitbucket per il tuo repository. Per ulteriori informazioni, consulta la sezione relativa all' esempio di richiesta pull Bitbucket per CodeBuild .	2 ottobre 2018
registri S3	CodeBuild ora supporta i log di compilazione in un bucket S3. In precedenza, era possibile creare registri solo utilizzando Logs. CloudWatch Per ulteriori informazioni, consulta la pagina relativa alla creazione di un progetto .	17 settembre 2018

[Molteplici sorgenti di input e più artefatti di output](#)

CodeBuild ora supporta progetti che utilizzano più di una fonte di input e pubblicano più di un set di artefatti. Per ulteriori informazioni, consulta [Esempio di sorgenti di input e artefatti di input multiple e di artefatti di input e CodePipeline integrazione con più sorgenti di input CodeBuild e artefatti di output.](#)

30 agosto 2018

[Esempio di versionamento semantico](#)

La Guida per CodeBuild l'utente contiene ora un esempio basato sui casi d'uso che dimostra come utilizzare il controllo delle versioni semantiche per creare nomi di artefatti in fase di compilazione. Per ulteriori informazioni, consulta [Utilizzo del controllo versioni semantico per nominare l'esempio di artefatti della compilazione.](#)

14 agosto 2018

[Nuovo esempio di sito Web statico](#)

La Guida per CodeBuild l'utente ora contiene un esempio basato sui casi d'uso che dimostra come ospitare l'output della build in un bucket S3. L'esempio sfrutta il recente supporto di artefatti di compilazione non crittografati. Per ulteriori informazioni, consulta [Creazione di un sito Web statico con output di compilazione ospitato in un bucket Amazon S3](#).

14 agosto 2018

[Support per sovrascrivere il nome di un artefatto con il controllo delle versioni semantiche](#)

È ora possibile utilizzare il controllo delle versioni semantiche per specificare un formato che utilizza per denominare gli artefatti di costruzione. CodeBuild Questo è utile perché un elemento di compilazione con un nome hardcoded sovrascrive elementi di compilazione precedenti che utilizzano lo stesso nome hardcoded. Ad esempio, se un processo di compilazione viene attivato più volte al giorno, puoi aggiungere un timestamp al nome dell'elemento. Il nome di ogni elemento di compilazione è univoco e non sovrascrive gli elementi dei processi di compilazione precedenti.

7 agosto 2018

[Support di artefatti di build non crittografati](#)

CodeBuild ora supporta build con artefatti di build non crittografati. Per ulteriori informazioni, consulta [Creare un progetto di compilazione](#) (console).

26 luglio 2018

[Support per CloudWatch parametri e allarmi Amazon](#)

CodeBuild ora fornisce l'integrazione con CloudWatch metriche e allarmi. Puoi utilizzare la CloudWatch console CodeBuild o per monitorare le build a livello di progetto e account. Per ulteriori informazioni, consulta la pagina relativa al [monitoraggio delle compilazioni](#).

19 luglio 2018

[Support per la segnalazione dello stato di una build](#)

CodeBuild ora puoi segnalare lo stato dell'inizio e del completamento di una build al tuo fornitore di origine. Per ulteriori informazioni, consulta [Creare un progetto di compilazione in CodeBuild](#).

10 luglio 2018

[Variabili di ambiente aggiunte alla CodeBuild documentazione](#)

La pagina [Variabili di ambiente in ambienti di compilazione](#) è stata aggiornata con le variabili di ambiente CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH e CODEBUILD_START_TIME.

9 luglio 2018

[Support per un finally blocco nel file buildspec](#)

La CodeBuild documentazione è stata aggiornata con dettagli sul finally blocco opzionale in un file buildspec . I comandi nel blocco finally vengono sempre eseguiti dopo i comandi nel blocco di comandi corrispondente. Per ulteriori informazioni, consulta [Sintassi Buildspec](#) .

20 giugno 2018

[CodeBuild notifiche di aggiornamento degli agenti](#)

La CodeBuild documentazione è stata aggiornata con dettagli su come utilizzare Amazon SNS per ricevere notifiche quando vengono rilasciate nuove versioni dell' CodeBuild agente. Per ulteriori informazioni, consulta [Ricevere notifiche per le nuove versioni degli AWS CodeBuild agenti](#).

15 giugno 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti apportate a ogni versione della Guida per l'utente di AWS CodeBuild prima di giugno 2018.

Modifica	Descrizione	Data
Supporto di build di Windows	CodeBuild ora supporta le build per la piattaforma Microsoft Windows Server, incluso un ambiente di compilazione preconfigurato per .NET Core 2.0	25 maggio 2018

Modifica	Descrizione	Data
	su Windows. Per ulteriori informazioni, consulta Esegui esempi di Microsoft Windows per CodeBuild .	
Supporto dell'idempotenza della compilazione	Quando esegui il <code>start-build</code> comando con AWS Command Line Interface (AWS CLI), puoi specificare che la build è idempotente. Per ulteriori informazioni, consulta Esecuzione di una compilazione (AWS CLI) .	15 maggio 2018
Supporto della sovrascrittura di più impostazioni di progetti di compilazione	Adesso, al momento della creazione di una compilazione, puoi sostituire più impostazioni del progetto di compilazione. Le sostituzioni riguardano solo questa compilazione. Per ulteriori informazioni, consulta Esegui AWS CodeBuild le build manualmente .	15 maggio 2018
Supporto dell'endpoint VPC	Adesso, puoi migliorare la sicurezza delle tue compilazioni utilizzando endpoint VPC. Per ulteriori informazioni, consulta Utilizzo degli endpoint VPC .	18 marzo 2018

Modifica	Descrizione	Data
Supporto di trigger	Adesso, puoi creare trigger per pianificare compilazioni a intervalli regolari. Per ulteriori informazioni, consulta Creare trigger AWS CodeBuild .	28 marzo 2018
Documentazione degli endpoint FIPS	Ora puoi imparare a usare AWS Command Line Interface (AWS CLI) o un AWS SDK per indicare di utilizzare uno dei quattro endpoint FIPS (CodeBuild Federal Information Processing Standards). Per ulteriori informazioni, consulta Specificare l' AWS CodeBuild endpoint .	28 marzo 2018
AWS CodeBuild disponibile in Asia Pacifico (Mumbai), Europa (Parigi) e Sud America (San Paolo)	AWS CodeBuild è ora disponibile nelle regioni Asia Pacifico (Mumbai), Europa (Parigi) e Sud America (San Paolo). Per ulteriori informazioni, consulta AWS CodeBuild nella Riferimenti generali di Amazon Web Services.	28 marzo 2018
GitHub Supporto per Enterprise Server	CodeBuild ora può creare a partire dal codice sorgente archiviato in un repository di GitHub Enterprise Server. Per ulteriori informazioni, consulta Esegui l'esempio di GitHub Enterprise Server .	25 gennaio 2018

Modifica	Descrizione	Data
Supporto di Git clone depth	CodeBuild ora supporta la creazione di un clone superficiale con una cronologia troncata al numero di commit specificato. Per ulteriori informazioni, consulta Creazione di un progetto di compilazione .	25 gennaio 2018
Supporto per VPC	Le compilazioni attivate da VPC ora possono accedere alle risorse all'interno del tuo VPC. Per ulteriori informazioni, consulta Supporto per VPC .	27 novembre 2017
Supporto del caching delle dipendenze	CodeBuild ora supporta la memorizzazione nella cache delle dipendenze. Ciò consente di CodeBuild salvare alcune parti riutilizzabili dell'ambiente di compilazione nella cache e di utilizzarle tra le build.	27 novembre 2017
Supporto di badge di compilazione	CodeBuild ora supporta l'uso dei build badge, che forniscono un'immagine incorporabile e generata dinamicamente (badge) che mostra lo stato dell'ultima build di un progetto. Per ulteriori informazioni, consulta Creazione di esempi di badge .	27 Novembre 2017

Modifica	Descrizione	Data
AWS Config integrazione	AWS Config ora supporta CodeBuild come AWS risorsa, il che significa che il servizio può tracciare i tuoi CodeBuild progetti. Per ulteriori informazioni su AWS Config, vedere AWS Config campione .	20 ottobre 2017
Ricostruisci automaticamente il codice sorgente aggiornato nei repository GitHub	Se il codice sorgente è archiviato in un GitHub repository, puoi abilitare la ricostruzione del codice sorgente ogni volta che una modifica AWS CodeBuild al codice viene inserita nel repository. Per ulteriori informazioni, consulta Esegui l' GitHub esempio di filtro pull request e webhook .	21 settembre 2017

Modifica	Descrizione	Data
Nuovi modi per archiviare e recuperare variabili di ambiente sensibili o di grandi dimensioni in Amazon EC2 Systems Manager Parameter Store	Ora puoi utilizzare la AWS CodeBuild console o il AWS CLI per recuperare variabili di ambiente sensibili o di grandi dimensioni archiviate in Amazon EC2 Systems Manager Parameter Store. Ora puoi anche utilizzare la AWS CodeBuild console per archiviare questi tipi di variabili di ambiente in Amazon EC2 Systems Manager Parameter Store. Precedentemente, potevi recuperare questi tipi di variabili di ambiente esclusivamente includendoli in una specifica di compilazione o eseguendo comandi di compilazione per automatizzare AWS CLI. Puoi memorizzare questi tipi di variabili di ambiente solo utilizzando la console Amazon EC2 Systems Manager Parameter Store. Per ulteriori informazioni, consulta Creazione di un progetto di compilazione, Modifica le impostazioni del progetto di compilazione, e Esegui le build manualmente.	14 settembre 2017

Modifica	Descrizione	Data
Supporto dell'eliminazione di compilazioni	Adesso, puoi eliminare compilazioni in AWS CodeBuild. Per ulteriori informazioni, consulta Eliminazione delle compilazioni .	31 agosto 2017
Modo aggiornato per recuperare variabili di ambiente sensibili o di grandi dimensioni archiviate in Amazon EC2 Systems Manager Parameter Store utilizzando un buildspec	AWS CodeBuild ora semplifica l'utilizzo di un buildspec per recuperare variabili di ambiente sensibili o di grandi dimensioni archiviate in Amazon EC2 Systems Manager Parameter Store. Precedentemente, potevi recuperare questi tipi di variabili di ambiente esclusivamente eseguendo comandi di compilazione per automatizzare l'AWS CLI. Per ulteriori informazioni, consulta la mappatura in. parameter-store Sintassi buildspec	10 agosto 2017
AWS CodeBuild supporta Bitbucket	CodeBuild ora può creare a partire dal codice sorgente memorizzato in un repository Bitbucket. Per ulteriori informazioni, consulta Creazione di un progetto di compilazione e Esegui le build manualmente .	10 agosto 2017

Modifica	Descrizione	Data
AWS CodeBuild disponibile negli Stati Uniti occidentali (California settentrionale), Europa (Londra) e Canada (Centrale)	AWS CodeBuild è ora disponibile nelle regioni Stati Uniti occidentali (California settentrionale), Europa (Londra) e Canada (Centrale). Per ulteriori informazioni, consulta AWS CodeBuild nella Riferimenti generali di Amazon Web Services.	29 giugno 2017
Posizioni e nomi di file di specifica di compilazione alternativi supportati	Adesso, puoi specificare una posizione o un nome di file alternativo per un file di specifica di build da utilizzare per un progetto di compilazione, anziché di un file di specifica di build predefinito denominato <code>buildspec.yml</code> alla radice del codice sorgente. Per ulteriori informazioni, consulta Nome del file buildspec e posizione di storage .	27 giugno 2017
Esempio di notifiche di compilazione aggiornato	CodeBuild ora offre supporto integrato per creare notifiche tramite Amazon CloudWatch Events e Amazon Simple Notification Service (Amazon SNS). Il precedente Esempio di notifiche di compilazione è stato aggiornato per mostrare questo nuovo funzionamento.	22 giugno 2017

Modifica	Descrizione	Data
Aggiunta di un esempio Docker in un'immagine personalizzata	È stato aggiunto un esempio che mostra come usare CodeBuild e un'immagine di build Docker personalizzata per creare ed eseguire un'immagine Docker. Per ulteriori informazioni, consulta la guida Docker in un esempio di immagine personalizzata .	7 giugno 2017
Recupera il codice sorgente per le richieste pull GitHub	Quando esegui una build CodeBuild che si basa sul codice sorgente archiviato in un GitHub repository, ora puoi specificare un ID di richiesta GitHub pull da compilare. Inoltre, puoi specificare un ID commit, un nome ramo o un nome tag. Per ulteriori informazioni, consulta il valore della versione di origine in Esecuzione di una compilazione (console) o il sourceVersion valore in. Esecuzione di una compilazione (AWS CLI)	6 giugno 2017

Modifica	Descrizione	Data
Versione della specifica di compilazione aggiornata	È stata rilasciata una nuova versione del formato di specifica di compilazione. La versione 0.2 risolve il problema dell' CodeBuild esecuzione di ogni comando build in un'istanza separata della shell predefinita. Inoltre nella versione 0.2 <code>environment_variables</code> viene rinominato in <code>env</code> e <code>plaintext</code> viene rinominato in <code>variables</code>. . Per ulteriori informazioni, consulta Riferimento alle specifiche di costruzione per CodeBuild.	9 maggio 2017
Dockerfiles per le immagini di compilazione disponibili in GitHub	Le definizioni per molte delle immagini di build fornite da AWS CodeBuild sono disponibili come Dockerfile in. GitHub Per ulteriori informazioni, consultate la colonna Definizione della tabella in. Immagini Docker fornite da CodeBuild	2 maggio 2017

Modifica	Descrizione	Data
AWS CodeBuild disponibile in Europa (Francoforte), Asia Pacifico (Singapore), Asia Pacifico (Sydney) e Asia Pacifico (Tokyo)	AWS CodeBuild è ora disponibile nelle regioni Europa (Francoforte), Asia Pacifico (Singapore), Asia Pacifico (Sydney) e Asia Pacifico (Tokyo). Per ulteriori informazioni, consulta AWS CodeBuild nella Riferimenti generali di Amazon Web Services.	21 marzo 2017
CodePipeline supporto per le azioni di test per CodeBuild	Ora puoi aggiungerlo a una pipeline in CodePipeline un'azione di test che utilizza CodeBuild. Per ulteriori informazioni, consulta Aggiungere un'azione CodeBuild di test a una pipeline (CodePipeline console) .	8 marzo 2017
Le specifiche di compilazione supportano il recupero di output di compilazione da directory di primo livello selezionate	I file Buildspec ora consentono di specificare singole directory di primo livello il cui contenuto è possibile indicare CodeBuild di includere negli artefatti di output della build. A tale scopo, devi utilizzare la mappatura base-directory . Per ulteriori informazioni, consulta Sintassi buildspec .	8 febbraio 2017

Modifica	Descrizione	Data
Variabili di ambiente integrate	AWS CodeBuild fornisce variabili di ambiente integrate aggiuntive da utilizzare nelle build. Queste includono variabili di ambiente che descrivono l'entità che ha avviato la compilazione, l'URL del repository del codice sorgente, l'ID versione del codice sorgente e altro ancora. Per ulteriori informazioni, consulta Variabili di ambiente degli ambienti di compilazione .	30 gennaio 2017
AWS CodeBuild disponibile negli Stati Uniti orientali (Ohio)	AWS CodeBuild è ora disponibile nella regione Stati Uniti orientali (Ohio). Per ulteriori informazioni, consulta AWS CodeBuild nella Riferimenti generali di Amazon Web Services.	19 gennaio 2017

Modifica	Descrizione	Data
Informazioni sul comportamento di shell e comando	CodeBuild esegue ogni comando specificato in un'istanza separata della shell predefinita di un ambiente di compilazione. Questo comportamento predefinito può produrre alcuni effetti collaterali imprevisti per i tuoi comandi. Consigliamo alcuni approcci alternativi per la risoluzione di questo comportamento predefinito, se necessario. Per ulteriori informazioni, consulta Shell e comandi negli ambienti di compilazione .	9 dicembre 2016
Informazioni sulle variabili di ambiente	CodeBuild fornisce diverse variabili di ambiente che è possibile utilizzare nei comandi di compilazione. Puoi inoltre definire le tue variabili di ambiente personali. Per ulteriori informazioni, consulta Variabili di ambiente degli ambienti di compilazione .	7 dicembre 2016
Argomento sulla risoluzione dei problemi	Ora sono disponibili informazioni relative alla risoluzione dei problemi. Per ulteriori informazioni, consulta Risoluzione dei problemi AWS CodeBuild .	5 dicembre 2016

Modifica	Descrizione	Data
Rilascio iniziale del plugin Jenkins	Questa è la versione iniziale del plugin CodeBuild Jenkins. Per ulteriori informazioni, consulta Uso AWS CodeBuild con Jenkins .	5 dicembre 2016
Versione iniziale della Guida per l'utente	Questa è la versione iniziale della Guida per l'utente di CodeBuild .	1° dicembre 2016

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.