



Guide de l'utilisateur

AWS CodeBuild



Version de l'API 2016-10-06

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeBuild: Guide de l'utilisateur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

| | |
|--|----|
| Qu'est-ce que AWS CodeBuild ? | 1 |
| | 1 |
| Comment courir CodeBuild | 1 |
| Tarification pour CodeBuild | 3 |
| Comment puis-je commencer CodeBuild ? | 3 |
| Concepts | 3 |
| Fonctionnement de CodeBuild | 3 |
| Étapes suivantes | 5 |
| Commencer | 6 |
| Mise en route à l'aide de la console | 6 |
| Étapes | 7 |
| Étape 1 : Création du code source | 7 |
| Étape 2 : Création du fichier buildspec | 10 |
| Étape 3 : Création de deux compartiments S3 | 12 |
| Étape 4 : Charger le code source et le fichier buildspec | 13 |
| Étape 5 : Créer le projet de génération | 15 |
| Étape 6 : Exécuter la génération | 17 |
| Étape 7 : Afficher les informations de génération résumées | 18 |
| Étape 8 : Afficher les informations de génération détaillées | 19 |
| Étape 9 : Obtenir l'artefact de sortie de génération | 20 |
| Étape 10 : Supprimer les compartiments S3 | 21 |
| Récapitulatif | 22 |
| Prise en main à l'aide de l'AWS CLI | 22 |
| Étapes | 23 |
| Étape 1 : Création du code source | 23 |
| Étape 2 : Création du fichier buildspec | 26 |
| Étape 3 : Création de deux compartiments S3 | 28 |
| Étape 4 : Charger le code source et le fichier buildspec | 29 |
| Étape 5 : Créer le projet de génération | 31 |
| Étape 6 : Exécuter la génération | 35 |
| Étape 7 : Afficher les informations de génération résumées | 37 |
| Étape 8 : Afficher les informations de génération détaillées | 40 |
| Étape 9 : Obtenir l'artefact de sortie de génération | 43 |
| Étape 10 : Supprimer les compartiments S3 | 44 |

| | |
|--|-----|
| Récapitulatif | 45 |
| Exemples | 46 |
| Exemples basés sur des cas d'utilisation | 46 |
| Échantillons multiservices | 47 |
| Exemple de badges de génération | 89 |
| Création d'un rapport de test à l'aide de l'exemple d' AWS CLI | 94 |
| Exemples de Docker pour CodeBuild | 101 |
| Hébergement d'une sortie de génération dans un compartiment S3 | 116 |
| Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie | 119 |
| Versions d'environnement d'exécution dans l'exemple de fichier buildspec | 123 |
| Exemple de version source | 132 |
| Exemples de référentiels sources tiers pour CodeBuild | 136 |
| Exemple d'utilisation de la gestion sémantique des versions pour nommer des artefacts de génération | 154 |
| Exemples Windows | 157 |
| Exécution des exemples | 157 |
| Structure de répertoire | 159 |
| Dépôt de | 159 |
| Planification d'une génération | 178 |
| Référence des spécifications de génération | 180 |
| Nom de fichier buildspec et emplacement de stockage | 181 |
| Syntaxe d'un fichier buildspec | 182 |
| Exemple de fichier buildspec | 204 |
| Versions de fichier buildspec | 208 |
| Référence des spécifications de génération par Batch | 208 |
| GitHubRéférence de spécification de construction d'Action Runner | 216 |
| Référence sur les environnements de génération | 218 |
| Images Docker fournies par CodeBuild | 219 |
| Modes et types de calcul de l'environnement de création | 239 |
| Shells et commandes dans les environnements de génération | 247 |
| Variables d'environnement dans les environnements de génération | 249 |
| Tâches en arrière-plan dans les environnements de génération | 254 |
| Construisez localement | 254 |
| Prérequis | 254 |
| Configurer l'image de construction | 255 |
| Exécutez leCodeBuildagent | 256 |

| | |
|--|-----|
| Réception de notifications en cas de nouvelle version de l'agent CodeBuild | 257 |
| Prise en charge de VPC | 259 |
| Cas d'utilisation | 259 |
| Autoriser l'accès à Amazon VPC dans votreCodeBuildprojets | 260 |
| Bonnes pratiques pour les VPC | 261 |
| Dépannage de la configuration de votre VPC | 262 |
| Limites des VPC | 263 |
| Utilisation de points de terminaison de VPC | 263 |
| Avant de créer des points de terminaison de VPC | 263 |
| Création des points de terminaison d'un VPC pour CodeBuild | 264 |
| Création d'une stratégie de point de terminaison de VPC pour CodeBuild | 265 |
| Modèle VPC AWS CloudFormation | 266 |
| Utilisation d'un serveur proxy | 271 |
| Composants requis pour exécuter CodeBuild sur un serveur proxy | 272 |
| Exécution de CodeBuild sur un serveur proxy explicite | 275 |
| Exécution de CodeBuild sur un serveur proxy transparent | 280 |
| Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy | 281 |
| Utilisation des projets de génération et des générations | 284 |
| Utilisation des projets de génération | 284 |
| Création d'un projet de génération | 285 |
| Création d'une règle de notification | 325 |
| Affichage d'une liste de noms de projet de génération | 328 |
| Affichage des détails d'un projet de génération | 331 |
| Mise en cache de génération | 334 |
| Créez des déclencheurs | 339 |
| Webhooks | 347 |
| Modification des paramètres d'un projet de génération | 376 |
| Suppression d'un projet de génération | 399 |
| Utilisation de projets partagés | 401 |
| Balisage d'un projet | 406 |
| Constructions par lots | 412 |
| GitHubAction Runner | 416 |
| Projets de build public | 424 |
| Utilisation des générations | 426 |
| Exécution d'une génération | 427 |
| Affichage des détails d'une génération | 439 |

| | |
|--|-----|
| Affichage d'une liste d'ID de génération | 441 |
| Affichage d'une liste d'ID de génération pour un projet de génération | 445 |
| Arrêt d'une génération | 449 |
| Arrêt d'une génération par lots | 450 |
| Retente une génération | 452 |
| Gestionnaire de session | 454 |
| Suppression de générations | 459 |
| Travailler avec le AWS Lambda calcul | 461 |
| Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda | 461 |
| Dans quelles régions le AWS Lambda calcul est pris en charge CodeBuild ? | 461 |
| Limites du AWS Lambda calcul | 462 |
| AWS Lambdacalculer des échantillons | 462 |
| Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java | 463 |
| Créer une application React d'une seule page avec CodeBuild Lambda Node.js | 467 |
| Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python | 470 |
| Travailler avec une capacité réservée | 475 |
| Comment démarrer avec les flottes à capacité réservée ? | 476 |
| Bonnes pratiques | 476 |
| Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ? | 477 |
| Quelles régions prennent en charge les flottes à capacité réservée ? | 477 |
| Propriétés du parc à capacité réservée | 477 |
| Échantillons de capacité réservée | 480 |
| Exemple de mise en cache avec capacité réservée | 480 |
| Limitations des flottes à capacité réservée | 482 |
| Utilisation des rapports de test | 483 |
| Créer un rapport de test | 484 |
| Utilisation de groupes de rapports | 485 |
| Créer un groupe de rapports | 487 |
| Mise à jour d'un groupe de rapports | 492 |
| Spécification des fichiers de test | 496 |
| Spécification des commandes de test | 496 |
| Attribution des noms des groupes de rapports | 497 |
| Baliser un groupe de rapports | 498 |
| Utilisation des groupes de rapports partagés. | 504 |
| Utilisation des rapports | 510 |

| | |
|---|-----|
| Utilisation des autorisations de rapport de test | 511 |
| Création d'un rôle pour les rapports de test | 512 |
| Autorisations pour les opérations de rapports de test | 514 |
| Exemples d'autorisations de rapport de test | 514 |
| Affichage des rapports de test | 515 |
| Affichage des rapports de test d'une génération | 515 |
| Affichage des rapports de test d'un groupe de rapports | 516 |
| Affichage des rapports de test dans votre compte AWS | 516 |
| Rapport de test avec des cadres de test | 516 |
| Génération de rapports avec Jasmine | 517 |
| Génération de rapports avec Jest | 519 |
| Génération de rapports avec pytest | 520 |
| Création de rapports avec RSpec | 521 |
| Rapports de couverture du code | 522 |
| | 522 |
| Création d'un rapport de couverture de code | 523 |
| Journalisation et surveillance | 525 |
| Journalisation des appels d'API AWS CodeBuild avec AWS CloudTrail | 525 |
| AWS CodeBuild informations dans CloudTrail | 525 |
| Présentation des AWS CodeBuild entrées des fichiers journaux | 526 |
| Surveillance des AWS CodeBuild | 529 |
| Métriques CloudWatch | 530 |
| métriques d'utilisation des ressources CloudWatch | 532 |
| Dimensions CloudWatch | 534 |
| Alarmes CloudWatch | 534 |
| Métriques CodeBuild | 535 |
| Métriques d'utilisation des ressources CodeBuild | 537 |
| Des alarmes CodeBuild | 541 |
| Sécurité | 543 |
| Protection des données | 543 |
| Chiffrement des données | 545 |
| Gestion des clés | 546 |
| Confidentialité du trafic | 546 |
| Gestion des identités et des accès | 547 |
| Présentation de la gestion des accès | 547 |
| Utilisation de politiques basées sur l'identité | 551 |

| | |
|--|-----|
| AWS CodeBuild référence aux autorisations | 580 |
| Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild | 587 |
| Affichage des ressources dans la console | 591 |
| Validation de conformité | 592 |
| Résilience | 593 |
| Sécurité de l'infrastructure | 593 |
| Accès au fournisseur de source | 594 |
| GitHub et jeton d'accès au serveur GitHub Enterprise | 594 |
| Mot de passe de l'application Bitbucket | 597 |
| Prévention du cas de figure de l'adjoint désorienté entre services | 601 |
| Rubriques avancées | 603 |
| Configuration avancée | 603 |
| Ajouter des autorisations d' CodeBuild accès à un groupe ou à un utilisateur IAM | 604 |
| Création d'un rôle CodeBuild de service | 611 |
| Création d'une clé gérée par le client | 619 |
| Installation et configuration de l'AWS CLI | 621 |
| Référence des commandes en ligne | 622 |
| Référence des kits SDK et des outils AWS | 623 |
| Kits SDK et outils AWS pris en charge pour AWS CodeBuild | 624 |
| Spécification du point de terminaison | 624 |
| Spécification du point de terminaison AWS CodeBuild (AWS CLI) | 625 |
| Spécification du point de terminaison AWS CodeBuild (kit SDK AWS) | 626 |
| À utiliser CodePipeline avec CodeBuild | 628 |
| Prérequis | 629 |
| Créer un pipeline (console) | 631 |
| Créer un pipeline (AWS CLI) | 636 |
| Action d'ajout d' | 641 |
| Action d'ajout d'essai | 645 |
| Utiliser CodeBuild avec Jenkins | 648 |
| Configuration de Jenkins | 648 |
| Installation du plug-in | 649 |
| Utilisation du plug-in | 649 |
| Utilisation d'CodeBuild Coecov | 651 |
| Intégration de Codecov dans un projet de génération | 651 |
| Applications sans serveur | 654 |
| Ressources connexes | 54 |

| | |
|---|-----|
| Résolution des problèmes | 656 |
| Apache Maven génère des artefacts de référence à partir du mauvais répertoire | 657 |
| Les commandes de génération s'exécutent en tant que racine par défaut | 659 |
| Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains Caractères anglais | 659 |
| Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store | 660 |
| Impossible d'accéder au filtre de branche dans la console CodeBuild | 661 |
| Impossible d'afficher la réussite ou l'échec de la génération | 662 |
| L'état du build n'a pas été communiqué au fournisseur source | 662 |
| Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019 | 662 |
| Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures | 663 |
| Erreur : accès refusé lors de la tentative de téléchargement du cache | 663 |
| Erreur : « BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE » lors de l'utilisation d'une image de génération personnalisée | 664 |
| Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces | 665 |
| Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération | 665 |
| Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction | 667 |
| Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl » | 668 |
| Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération | 668 |
| Erreur : « Git clone failed: Unable to access 'your-repository-URL': SSL certificate problem: Self signed certificate » | 668 |
| Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération | 669 |
| Erreur : « This build image requires selecting at least one runtime version. » | 670 |
| Erreur : « QUEUED: INSUFFICIENT_SUBNET » lorsqu'une génération dans une file d'attente échoue | 671 |
| Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie » | 671 |

| | |
|---|---------|
| Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied» | 672 |
| Erreur : « Unable to locate credentials » | 672 |
| RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy | 674 |
| Le shell bourne (sh) doit exister dans les images de génération | 675 |
| Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération | 676 |
| Erreur : « Impossible de vérifier JobWorker l'identité » | 676 |
| La construction n'a pas pu démarrer | 676 |
| Accès aux GitHub métadonnées dans les versions mises en cache localement | 677 |
| AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3... | 677 |
| Quotas | 678 |
| Quotas de service | 678 |
| Autres limites | 683 |
| Projets de génération | 683 |
| Versions | 684 |
| Flottes informatiques | 684 |
| Rapports | 686 |
| Balises | 686 |
| Notices de tiers AWS CodeBuild pour Windows | 688 |
| 1) image Docker de base — Windowsservercore | 688 |
| 2) Image Docker basée sur Windows : choco | 690 |
| 3) Image Docker basée sur Windows : git --version 2.16.2 | 690 |
| 4) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools | 690 |
| 5) Image Docker basée sur Windows : nuget.commandline --version 4.5.1 | 695 |
| 7) Image Docker basée sur Windows : netfx-4.6.2-devpack | 695 |
| 8) Image Docker basée sur Windows — visualfsharptools, version 4.0 | 697 |
| 9) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies | 697 |
| 10) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1 | 701 |
| 11) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx | 705 |
| 12) Image Docker basée sur Windows : dotnet-sdk | 707 |
| Historique de la documentation | 708 |
| Mises à jour antérieures | 727 |
| Glossaire AWS | 743 |
| | dccxliv |

Qu'est-ce que AWS CodeBuild ?

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

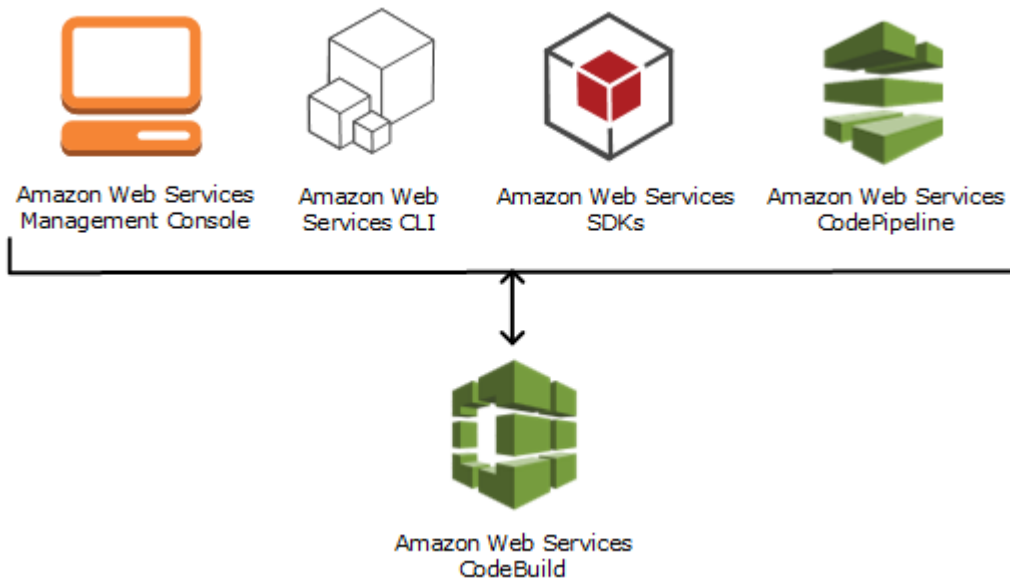
CodeBuild fournit les avantages suivants :

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

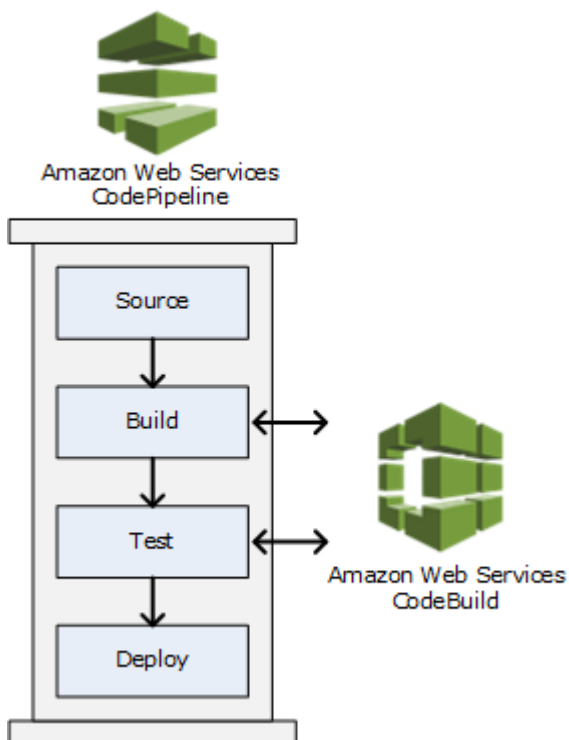
Pour plus d'informations, consultez [AWS CodeBuild](#).

Comment courir CodeBuild

Vous pouvez utiliser la console AWS CodeBuild ou AWS CodePipeline pour exécuter CodeBuild. Vous pouvez également automatiser l'exécution de CodeBuild en utilisant le AWS Command Line Interface (AWS CLI) ou les AWS SDK.



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



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

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

Tarification pour CodeBuild

Pour plus d'informations, consultez [CodeBuild les tarifs](#).

Comment puis-je commencer CodeBuild ?

Nous vous recommandons d'effectuer les étapes suivantes :

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

Concepts AWS CodeBuild

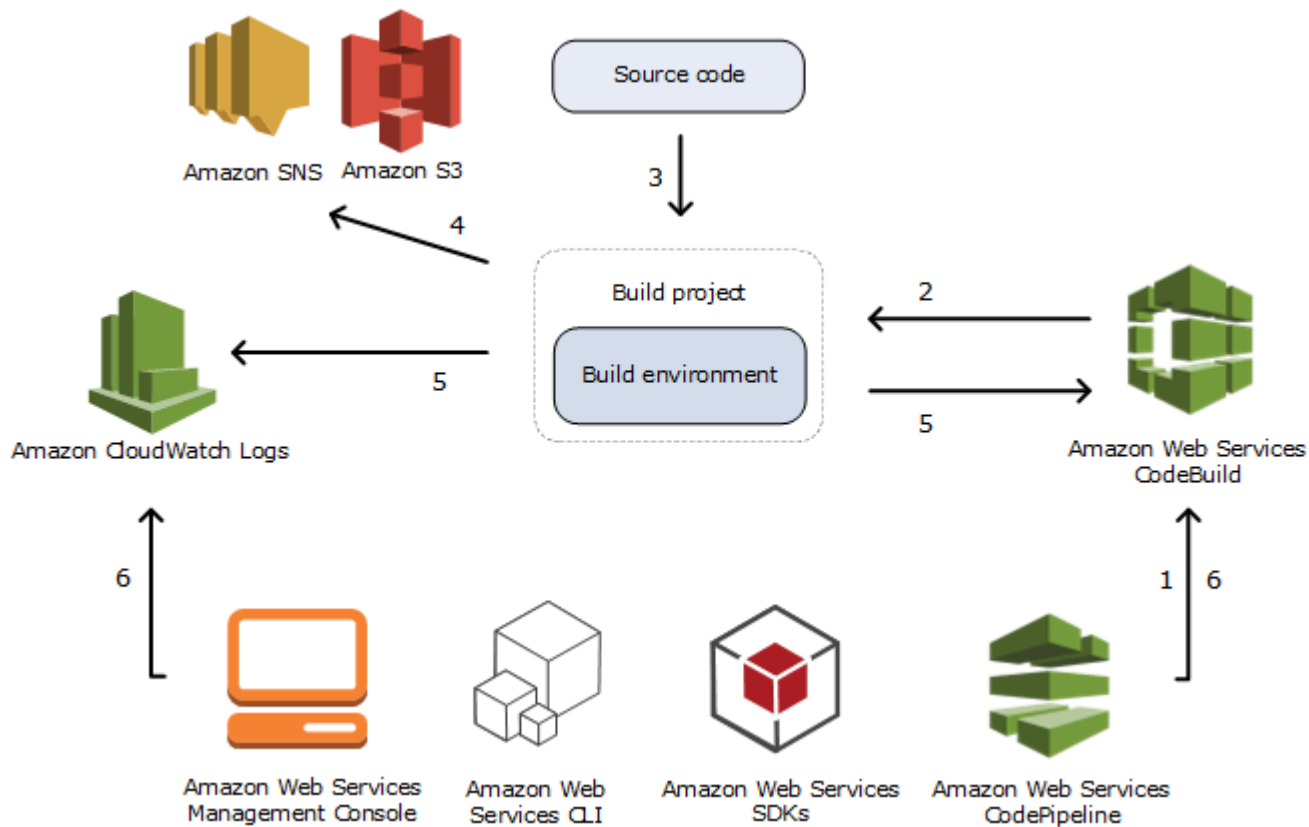
Les concepts suivants sont importants pour comprendre le fonctionnement de CodeBuild.

Rubriques

- [Fonctionnement de CodeBuild](#)
- [Étapes suivantes](#)

Fonctionnement de CodeBuild

Le schéma suivant montre ce qui se produit lorsque vous exécutez une génération avec CodeBuild :



- Comme entrée, vous devez fournir à CodeBuild un projet de génération. UNProjet de générationinclut des informations sur la procédure d'exécution d'une génération, y compris l'emplacement où obtenir le code source, l'environnement de génération à utiliser, les commandes de génération à exécuter et l'emplacement où stocker la sortie de génération. UNEnvironnement de générationreprésente une combinaison de système d'exploitation, d'environnement d'exécution de langage de programmation et d'outils que CodeBuild utilise pour exécuter une génération. Pour plus d'informations, consultez :
 - [Création d'un projet de génération](#)
 - [Référence sur les environnements de génération](#)
- CodeBuild utilise le projet de génération pour créer l'environnement de génération.
- CodeBuildtélécharge le code source dans l'environnement de génération, puis utilise la spécification de build (buildspec), telle que définie dans le projet de build ou incluse directement dans le code source. UNBuildspecest un ensemble de commandes de génération et de paramètres associés, au format YAML, que CodeBuild utilise pour exécuter une génération. Pour plus d'informations, consultez l'[Référence des spécifications de génération](#).

4. S'il existe une sortie de génération, l'environnement de génération charge sa sortie dans un compartiment S3. L'environnement de génération peut également effectuer les tâches que vous spécifiez dans la spécification de génération (par exemple, envoyer des notifications de génération vers une rubrique Amazon SNS). Pour voir un exemple, consultez [Exemple de notifications de génération](#).
5. Pendant que la génération s'exécute, l'environnement de génération envoie des informations vers CodeBuildWatch Amazon CloudWatch Logs.
6. Pendant que la génération s'exécute, vous pouvez utiliser le paramètre `AWS CodeBuildconsole` AWS CLI, ou `AWSKit` SDK pour obtenir des informations de génération résumées auprès de CodeBuild et des informations de génération détaillées auprès d'Amazon CloudWatch Logs. Si vous utilisez `AWS CodePipeline` pour exécuter des générations, vous pouvez obtenir des informations de génération limitées auprès de CodePipeline.

Étapes suivantes

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

1. Expérimenter avec CodeBuild dans un exemple de scénario en suivant les instructions de [Mise en route à l'aide de la console](#).
2. Utiliser CodeBuild dans vos propres scénarios en suivant les instructions de [Planification d'une génération](#).

Mise en route d'sur CodeBuild

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

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

Important

Il n'est pas recommandé d'utiliser votre compte AWS racine pour suivre ce didacticiel.

Mise en route d'AWS CodeBuild à l'aide de la console

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

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

Important

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

Étapes

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

Étape 1 : Création du code source

(Partie de : [Mise en route d'AWS CodeBuild à l'aide de la console](#))

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

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

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

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

```
public class MessageUtil {
    private String message;
```

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

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

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

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

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

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

public class TestMessageUtil {

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

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

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

```
}  
}
```

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

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

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

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

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

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

Étape suivante

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

Étape 2 : Création du fichier buildspec

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

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

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

```
version: 0.2

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

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

Important

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

Note

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

Dans cette déclaration de spécification de génération :

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

Dans cet exemple, pendant la `build` phase, CodeBuild exécute la `mvn install` commande. Cette commande demande à Apache Maven de compiler, tester et packager les fichiers de classe Java compilés dans un artefact de sortie de génération. A des fins d'exhaustivité, quelques `echo`

commandes sont placées dans chaque phase de génération de cet exemple. Lorsque vous afficherez les informations détaillées de génération ultérieurement dans ce didacticiel, la sortie de ces commandes echo vous aidera à mieux comprendre comment CodeBuild exécute les commandes et dans quel ordre. (Même si toutes les phases de génération sont incluses dans cet exemple, vous n'êtes pas obligé d'inclure une phase de génération si vous n'avez pas l'intention de lancer des commandes au cours de cette phase.) Pour chaque phase de construction, CodeBuild exécute chaque commande spécifiée, une par une, dans l'ordre indiqué, du début à la fin.

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

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

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

Étape suivante

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

Étape 3 : Création de deux compartiments S3

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

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

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

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

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

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

Note

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

Étape suivante

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

Étape 4 : Charger le code source et le fichier buildspec

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

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

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

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

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

Important

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

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

Important

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

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

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

Étape suivante

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

Étape 5 : Créer le projet de génération

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

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

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

Pour créer le projet de génération

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Utilisez le sélecteur de AWS région pour choisir une AWS région prise en charge. CodeBuild Pour plus d'informations, consultez [Points de terminaison et quotas AWS CodeBuild](#) dans le document Référence générale d'Amazon Web Services.
3. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

4. Sur la page Create build project (Créer un projet de génération), dans Project configuration (Configuration du projet), pour Nom du projet, saisissez un nom pour ce projet de génération (dans cet exemple, codebuild-demo-project). Les noms de projet de génération doivent être uniques dans chaque compte AWS. Si vous utilisez un autre nom, veillez à l'utiliser tout au long de ce didacticiel.

 Note

Sur la page Create build project (Créer un projet de génération), un message d'erreur similaire à ce qui suit peut s'afficher : You are not authorized to perform this operation (Vous n'êtes pas autorisé à effectuer cette opération). Cela est probablement dû au fait que vous vous êtes connecté au en AWS Management Console tant qu'utilisateur qui n'est pas autorisé à créer un projet de construction. Pour résoudre ce problème, déconnectez-vous de AWS Management Console, puis reconnectez-vous avec les informations d'identification appartenant à l'une des entités IAM suivantes :

- Utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur et de votre premier groupe Compte AWS root](#) dans le guide de l'utilisateur.
- Un utilisateur de votre AWS compte dont les politiques `AWSCodeBuildAdminAccessAmazonS3ReadOnlyAccess`, et `IAMFullAccess` gérées sont associées à cet utilisateur ou à un groupe IAM auquel appartient l'utilisateur. Si aucun utilisateur ou groupe de votre AWS compte ne possède ces autorisations et que vous ne pouvez pas les ajouter à votre utilisateur ou à votre groupe, contactez l'administrateur de votre AWS compte pour obtenir de l'aide. Pour plus d'informations, consultez [AWS politiques gérées \(prédéfinies\) pour AWS CodeBuild](#).

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

5. Dans Source, dans Source provider, choisissez Amazon S3.
6. **Pour Bucket, choisissez `codebuild-Region-ID - Account-ID -input-bucket`.**

7. Pour S3 object key (Clé d'objet S3), saisissez **MessageUtil.zip**.
8. Dans Environnement, pour Image d'environnement, laissez Image gérée sélectionné.
9. Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
10. Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
11. Pour Image, choisissez aws/codebuild/amazonlinux2-x86_64-standard:4.0.
12. Dans Rôle de service, laissez Nouveau rôle de service sélectionné et laissez Nom du rôle inchangé.
13. Pour Buildspec, laissez Utiliser un fichier buildspec sélectionné.
14. Dans Artifacts, pour Type, choisissez Amazon S3.
15. *Pour le nom du compartiment, choisissez **codebuild- Region-ID - Account-ID -output-bucket**.*
16. Laissez Nom et Chemin d'accès vides.
17. Choisissez Créer un projet de génération.

Étape suivante

[Étape 6 : Exécuter la génération](#)

Étape 6 : Exécuter la génération

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

Au cours de cette étape, vous demandez à AWS CodeBuild d'exécuter la génération avec les paramètres du projet de génération.

Pour exécuter la génération

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de construction, choisissez codebuild-demo-project, puis choisissez Démarrer la construction. La construction démarre immédiatement.

Étape suivante

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

Étape 7 : Afficher les informations de génération résumées

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

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

Pour afficher les informations de génération résumées

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

Dans État de la génération, Réussi doit être affiché.

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

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

Étape suivante

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

Étape 8 : Afficher les informations de génération détaillées

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

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

Note

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

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

Pour afficher les informations de génération détaillées

1. Avec la page de détails de génération toujours affichée de l'étape précédente, les 10,000 dernières lignes du journal de génération apparaissent dans Journaux de génération. Pour voir l'intégralité du journal de construction dans CloudWatch Logs, cliquez sur le lien [Afficher le journal complet](#).
2. Dans le flux du journal CloudWatch des journaux, vous pouvez parcourir les événements du journal. Par défaut, seul le dernier ensemble d'événements de journal s'affiche. Pour afficher des événements de journal précédents, faites défiler vers la liste.
3. Dans ce didacticiel, la plupart des événements de journal contiennent des informations détaillées sur le téléchargement et l'installation des fichiers de dépendance de génération par CodeBuild

dans son environnement de génération, ce qui ne vous intéresse probablement pas. Vous pouvez utiliser la zone Filtrer les événements pour limiter les informations affichées. Par exemple, si vous entrez "[INFO]" dans Filtrer les événements, seuls les événements contenant [INFO] s'affichent. Pour plus d'informations, consultez la section [Syntaxe des filtres et des modèles](#) dans le guide de CloudWatch l'utilisateur Amazon.

Étape suivante

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

Étape 9 : Obtenir l'artefact de sortie de génération

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

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

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

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

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

Note

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

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

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

1. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
2. Ouvrir codebuild-*region-ID-account-ID*-output-bucket.

3. Ouvrez le dossier `codebuild-demo-project`.
4. Ouvrez le dossier `target`, dans lequel se trouve le fichier d'artefact de sortie de génération `messageUtil-1.0.jar`.

Étape suivante

[Étape 10 : Supprimer les compartiments S3](#)

Étape 10 : Supprimer les compartiments S3

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

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

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

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

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

```
}
```

Étape suivante

[Récapitulatif](#)

Récapitulatif

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

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

Prise en main d'AWS CodeBuild à l'aide de l'AWS CLI

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

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

Important

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

Étapes

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

Étape 1 : Création du code source

(Partie de : [Prise en main d'AWS CodeBuild à l'aide de l'AWS CLI](#))

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

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

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

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

```
public class MessageUtil {
    private String message;
```

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

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

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

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

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

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

public class TestMessageUtil {

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

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

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

```
}  
}
```

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

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

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

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

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

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

Étape suivante

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

Étape 2 : Création du fichier buildspec

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

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

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

```
version: 0.2

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

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

Important

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

Note

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

Dans cette déclaration de spécification de génération :

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

Dans cet exemple, pendant la `build` phase, CodeBuild exécute la `mvn install` commande. Cette commande demande à Apache Maven de compiler, tester et packager les fichiers de classe Java compilés dans un artefact de sortie de génération. A des fins d'exhaustivité, quelques `echo`

commandes sont placées dans chaque phase de génération de cet exemple. Lorsque vous afficherez les informations détaillées de génération ultérieurement dans ce didacticiel, la sortie de ces commandes echo vous aidera à mieux comprendre comment CodeBuild exécute les commandes et dans quel ordre. (Même si toutes les phases de génération sont incluses dans cet exemple, vous n'êtes pas obligé d'inclure une phase de génération si vous n'avez pas l'intention de lancer des commandes au cours de cette phase.) Pour chaque phase de construction, CodeBuild exécute chaque commande spécifiée, une par une, dans l'ordre indiqué, du début à la fin.

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

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

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

Étape suivante

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

Étape 3 : Création de deux compartiments S3

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

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

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

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

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

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

Note

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

Étape suivante

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

Étape 4 : Charger le code source et le fichier buildspec

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

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

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

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

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

Important

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

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

Important

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

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

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

Étape suivante

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

Étape 5 : Créer le projet de génération

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

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

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

Pour créer le projet de génération

1. Utilisez l'interface AWS CLI pour exécuter la commande create-project :

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

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

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

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

Remplacez *ServiceIamRole* par le nom de ressource Amazon (ARN) d'un rôle CodeBuild de service (par exemple,). `arn:aws:iam::account-ID:role/role-name` Pour en créer un, consultez [Création d'un rôle CodeBuild de service](#).

Dans ces données :

- `name` représente un identificateur obligatoire pour ce projet de génération (dans cet exemple, `codebuild-demo-project`). Les noms de projet de génération doivent être uniques au sein de votre compte.
- Pour `source`, `type` est une valeur obligatoire qui représente le type de référentiel du code source (dans cet exemple, `S3` pour un compartiment Amazon S3).
- Pour `source`, `location` représente le chemin d'accès au code source (dans cet exemple, le nom du compartiment d'entrée, suivi du nom du fichier ZIP).
- Pour `artifacts`, `type` est une valeur obligatoire qui représente le type de référentiel de l'artefact de sortie de construction (dans cet exemple, `S3` pour un compartiment Amazon S3).
- Pour `artifacts`, `location` représente le nom du compartiment de sortie que vous avez créé ou identifié précédemment (dans cet exemple, `codebuild-region-ID-account-ID-output-bucket`).

- Pour `environment`, `type` est une valeur obligatoire qui représente le type d'environnement de construction (dans cet exemple, `LINUX_CONTAINER`).
- Pour `environment`, `image` est une valeur obligatoire qui représente le nom de l'image Docker et la combinaison de balises utilisés par ce projet de génération, telle que spécifiée par le type de référentiel d'images Docker (dans cet exemple, `aws/codebuild/standard:5.0` pour une image Docker dans le référentiel d'images CodeBuild Docker). `aws/codebuild/standard` est le nom de l'image Docker. `5.0` est le tag de l'image Docker.

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

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

Note

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

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

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

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

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
```

```
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "name": "message-util.zip"
  },
  "lastModified": 1472661575.244,
  "timeoutInMinutes": 60,
  "created": 1472661575.244,
  "environment": {
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/standard:5.0",
    "type": "LINUX_CONTAINER",
    "environmentVariables": []
  },
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
  "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
}
}
```

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

- `arn` représente l'ARN du projet de génération.

Note

Après avoir exécuté la `create-project` commande, un message d'erreur similaire au suivant peut s'afficher : Utilisateur : **User-ARN** n'est pas autorisé à exécuter : `codebuild :. CreateProject` Cela est probablement dû au fait que vous avez configuré le AWS CLI avec les informations d'identification d'un utilisateur qui ne dispose pas des autorisations suffisantes CodeBuild pour créer des projets de construction. Pour résoudre ce problème, configurez les informations AWS CLI d'identification appartenant à l'une des entités IAM suivantes :

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

Étape suivante

[Étape 6 : Exécuter la génération](#)

Étape 6 : Exécuter la génération

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

Au cours de cette étape, vous demandez à AWS CodeBuild d'exécuter la génération avec les paramètres du projet de génération.

Pour exécuter la génération

1. Utilisez l'interface AWS CLI pour exécuter la commande `start-build` :

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

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

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

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

- build représente les informations sur cette génération.
 - buildComplete indique si la génération a été terminée (true). Sinon la valeur est renvoyé, false.
 - initiator représente l'entité qui a démarré la génération.

- `artifacts` représente les informations sur la sortie de génération, y compris son emplacement.
- `projectName` représente le nom du projet de génération.
- `buildStatus` représente l'état actuel de la génération lorsque la commande `start-build` a été exécutée.
- `currentPhase` représente la phase actuelle de génération lorsque la commande `start-build` a été exécutée.
- `startTime` représente l'heure, au format horaire Unix, à laquelle le processus de génération a démarré.
- `id` représente l'ID de la génération.
- `arn` représente l'ARN de la génération.

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

Étape suivante

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

Étape 7 : Afficher les informations de génération résumées

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

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

Pour afficher les informations de génération résumées

Utilisez l'interface AWS CLI pour exécuter la commande `batch-get-builds`.

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

Remplacez `id` par la valeur `id` qui apparaît dans la sortie de l'étape précédente.

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

```
{
  "buildsNotFound": [],
  "builds": [
    {
```

```
"buildComplete": true,
"phases": [
  {
    "phaseStatus": "SUCCEEDED",
    "endTime": 1472848788.525,
    "phaseType": "SUBMITTED",
    "durationInSeconds": 0,
    "startTime": 1472848787.882
  },
  ... The full list of build phases has been omitted for brevity ...
  {
    "phaseType": "COMPLETED",
    "startTime": 1472848878.079
  }
],
"logs": {
  "groupName": "/aws/codebuild/codebuild-demo-project",
  "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
  "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
},
"artifacts": {
  "md5sum": "MD5-hash",
  "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/message-util.zip",
  "sha256sum": "SHA-256-hash"
},
"projectName": "codebuild-demo-project",
"timeoutInMinutes": 60,
"initiator": "user-name",
"buildStatus": "SUCCEEDED",
"environment": {
  "computeType": "BUILD_GENERAL1_SMALL",
  "image": "aws/codebuild/standard:5.0",
  "type": "LINUX_CONTAINER",
  "environmentVariables": []
},
"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"currentPhase": "COMPLETED",
"startTime": 1472848787.882,
```



```
    "endTime": 1472848878.079,  
    "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",  
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-  
project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"  
  }  
]  
}
```

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

Note

Vous pouvez également utiliser la console Amazon S3 pour afficher ces hachages. Cochez la case en regard de l'artefact de sortie de génération, choisissez Actions, puis choisissez Propriétés. Dans le volet Propriétés, développez Metadata et visualisez les valeurs de `x-amz-meta-codebuild-content-md5` et `-content-sha256`. `x-amz-meta-`

codebuild (Dans la console Amazon S3, la valeur ETag de l'artefact de sortie de compilation ne doit pas être interprétée comme étant le hachage MD5 ou SHA-256.) Si vous utilisez les kits SDK AWS pour obtenir ces hachages, les valeurs sont nommées `codebuild-content-md5` et `codebuild-content-sha256`.

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

Note

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

Étape suivante

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

Étape 8 : Afficher les informations de génération détaillées

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

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

Note

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

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

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

Pour afficher les informations de génération détaillées

1. Utilisez votre navigateur Web pour accéder à l'emplacement deepLink qui apparaît dans la sortie de l'étape précédente (par exemple, `https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).
2. Dans le flux du journal CloudWatch des journaux, vous pouvez parcourir les événements du journal. Par défaut, seul le dernier ensemble d'événements de journal s'affiche. Pour afficher des événements de journal précédents, faites défiler vers la liste.
3. Dans ce didacticiel, la plupart des événements de journal contiennent des informations détaillées sur le téléchargement et l'installation des fichiers de dépendance de génération par CodeBuild dans son environnement de génération, ce qui ne vous intéresse probablement pas. Vous pouvez utiliser la zone Filtrer les événements pour limiter les informations affichées. Par exemple, si vous entrez "[INFO]" dans Filtrer les événements, seuls les événements contenant [INFO] s'affichent. Pour plus d'informations, consultez la section [Syntaxe des filtres et des modèles](#) dans le guide de CloudWatch l'utilisateur Amazon.

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

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
[Container] 2016/04/15 17:49:44 [INFO]
[Container] 2016/04/15 17:49:44 [INFO]
-----
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
-----
```

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

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

Étape suivante

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

Étape 9 : Obtenir l'artefact de sortie de génération

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

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

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

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

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

Note

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

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

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

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

Étape suivante

[Étape 10 : Supprimer les compartiments S3](#)

Étape 10 : Supprimer les compartiments S3

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

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

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

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

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Étape suivante

[Récapitulatif](#)

Récapitulatif

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

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

CodeBuild échantillons

Ces groupes d'échantillons peuvent être utilisés pour expérimenter avec AWS CodeBuild :

Rubriques

- [Utilisez des exemples basés sur des cas pour CodeBuild](#)
- [Exemples Microsoft Windows pour CodeBuild](#)

Utilisez des exemples basés sur des cas pour CodeBuild

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

[Échantillons multiservices](#)

Une liste d'échantillons multiservices à tester. AWS CodeBuild

[Exemple de badges de génération](#)

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

[Création d'un rapport de test à l'aide de l'exemple d' AWS CLI](#)

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

[Exemples de Docker pour CodeBuild](#)

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

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

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

[Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#)

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

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

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

[Exemple de version source](#)

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

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

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

[Exemple d'utilisation de la gestion sémantique des versions pour nommer des artefacts de génération](#)

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

Exemples multiservices pour CodeBuild

Vous pouvez utiliser ces exemples multiservices pour tester : AWS CodeBuild

[Échantillon Amazon ECR](#)

Utilise une image Docker dans un référentiel Amazon ECR pour utiliser Apache Maven afin de produire un seul fichier JAR.

[Exemple Amazon EFS](#)

Montre comment configurer un fichier buildspec afin qu'un CodeBuild projet soit monté et construit sur un système de fichiers Amazon EFS.

[AWS CodePipeline échantillons](#)

Montre comment AWS CodePipeline créer un build avec des builds par lots ainsi que plusieurs sources d'entrée et plusieurs artefacts de sortie.

[AWS Config échantillon](#)

Montre comment procéder à la configuration AWS Config. Répertorie les CodeBuild ressources qui font l'objet d'un suivi et décrit comment rechercher CodeBuild des projets AWS Config.

[Exemple de notifications de génération](#)

Utilise Apache Maven pour produire un fichier JAR unique. Envoie une notification de création aux abonnés d'une rubrique Amazon SNS.

Exemple Amazon ECR pour CodeBuild

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

Important

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

Exécution de l'exemple

Pour exécuter cet exemple

1. Pour créer et transférer l'image Docker vers votre référentiel d'images dans Amazon ECR, suivez les étapes décrites dans la section « Exécution de l'exemple » du [Publier une image Docker dans un exemple de référentiel d'images Amazon ECR](#)
2. Créer un projet Go :
 - a. Créez les fichiers comme décrit dans les [Fichiers d'un projet Go](#) sections [Structure d'un projet Go](#) et de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 ou un AWS CodeCommit référentiel Bitbucket. GitHub

Important

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

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

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

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

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

- c. Pour obtenir le résultat de l'artefact de sortie de la génération, ouvrez votre compartiment de sortie S3.
 - d. Téléchargez le fichier `GoOutputArtifact.zip` sur votre ordinateur ou instance local, puis extrayez le contenu du fichier. Dans le contenu extrait, obtenez le fichier `hello`.
3. Si l'une des conditions suivantes est vraie, vous devez ajouter des autorisations à votre référentiel d'images dans Amazon ECR afin de AWS CodeBuild pouvoir intégrer son image Docker dans l'environnement de génération.
- Votre projet utilise des CodeBuild informations d'identification pour extraire les images Amazon ECR. Ceci est indiqué par la valeur `CODEBUILD` dans l'attribut `imagePullCredentialsType` de votre `ProjectEnvironment`.
 - Votre projet utilise une image Amazon ECR multi-comptes. Dans ce cas, votre projet doit utiliser son rôle de service pour extraire les images Amazon ECR. Pour activer

ce comportement, définissez l'attribut `imagePullCredentialsType` de votre `ProjectEnvironment` sur `SERVICE_ROLE`.

1. Ouvrez la console Amazon ECR à l'adresse <https://console.aws.amazon.com/ecr/>.
2. Dans la liste des noms de référentiel, choisissez le nom du référentiel que vous avez créé ou sélectionné.
3. Dans le volet de navigation, choisissez Permissions (Autorisations), Edit (Modifier), puis Add statement (Ajouter une instruction).
4. Pour Statement name, saisissez un identifiant (par exemple, **CodeBuildAccess**).
5. Pour Effect (Effet), conservez la sélection Allow (Autoriser). Cela indique que vous souhaitez autoriser l'accès à un autre compte AWS .
6. Pour Principal, effectuez l'une des opérations suivantes:
 - Si votre projet utilise des CodeBuild informations d'identification pour extraire une image Amazon ECR, dans Service principal, entrez **codebuild.amazonaws.com**.
 - Si votre projet utilise une image Amazon ECR multi-comptes, saisissez les identifiants des AWS AWS comptes auxquels vous souhaitez donner accès pour les identifiants de compte.
7. Ignorer la liste Toutes les entités IAM.
8. Pour Action, sélectionnez les actions à extraction uniquement : `ecr :GetDownloadUrlForLayer`, `ecr :` et `ecr :BatchGetImage`. `BatchCheckLayerAvailability`
9. Pour Conditions, ajoutez ce qui suit :

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

10. Choisissez Enregistrer.

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

- Si votre projet utilise des CodeBuild informations d'identification pour extraire une image Amazon ECR, cela `"codebuild.amazonaws.com"` apparaît sous Principaux du service.

- Si votre projet utilise une image Amazon ECR multi-comptes, l'identifiant du AWS compte auquel vous souhaitez donner accès apparaît sous Identifiants de AWS compte.

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:<region>:<aws-account-id>:project/<project-name>",
          "aws:SourceAccount": "<aws-account-id>"
        }
      }
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<AWS-account-ID>:root"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    }
  ]
}
```

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<AWS-account-ID>:root"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    }
  ]
}
```

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

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

```
{
  "name": "amazon-ecr-sample-project",
```

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

5. Pour obtenir le résultat de l'artefact de sortie de la génération, ouvrez votre compartiment de sortie S3.
6. Téléchargez le fichier *GoOutputArtifact*.zip sur votre ordinateur ou instance local, puis extrayez le contenu du fichier *GoOutputArtifact*.zip. Dans le contenu extrait, obtenez le fichier hello.

Structure d'un projet Go

Cet exemple suppose que cette structure de répertoire est utilisée.

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

Fichiers d'un projet Go

Cet exemple utilise ces fichiers.

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

```
version: 0.2
```

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

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

```
package main
import "fmt"

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

Ressources connexes

- Pour plus d'informations sur la prise en AWS CodeBuild main, consultez [Mise en route d'AWS CodeBuild à l'aide de la console](#).
- Pour plus d'informations sur la résolution des problèmes dans CodeBuild, voir [Résolution des problèmes AWS CodeBuild](#).
- Pour plus d'informations sur les quotas dans CodeBuild, voir [Quotas pour AWS CodeBuild](#).

Exemple d'Amazon Elastic File System pour AWS CodeBuild

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

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

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

Étapes de haut niveau

Cet exemple couvre les trois étapes de haut niveau requises pour utiliser Amazon EFS avec AWS CodeBuild :

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

Note

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

Créez un VPC à l'aide de AWS CloudFormation

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

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

Note

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

2. [Connectez-vous à la console Amazon VPC AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/vpc/.](https://console.aws.amazon.com/vpc/)
3. Choisissez le VPC avec lequel vous avez créé. AWS CloudFormation
4. Dans l'onglet Description, notez le nom de votre VPC et son ID. Les deux sont obligatoires lorsque vous créez votre AWS CodeBuild projet ultérieurement dans cet exemple.

Créez un système de fichiers Amazon Elastic File System avec votre VPC

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

1. Connectez-vous à la console Amazon EFS AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/efs/.](https://console.aws.amazon.com/efs/)
2. Choisissez Create file system (Créer un système de fichiers).
3. Dans VPC, choisissez le nom de VPC que vous avez noté précédemment dans cet exemple.
4. Conservez les Zones de disponibilité associées à vos sous-réseaux sélectionnés.


5. Choisissez Étape suivante.
6. Dans Ajouter des balises, pour la clé Nom par défaut, dans Valeur, entrez le nom de votre système de fichiers Amazon EFS.
7. Conservez Bursting (Transmission en rafales) et General Purpose (Usage général) sélectionnés en tant que modes de performance et de débit par défaut, puis choisissez Étape suivante.
8. Pour Configurer client access (Configurer l'accès client), choisissez Next Step (Étape suivante).
9. Sélectionnez Créer un système de fichiers.
10. (Facultatif) Nous vous recommandons d'ajouter une politique à votre système de fichiers Amazon EFS qui applique le chiffrement des données en transit. Dans la console Amazon EFS, choisissez la politique du système de fichiers, choisissez Modifier, cochez la case intitulée Appliquer le chiffrement en transit pour tous les clients, puis sélectionnez Enregistrer.

Création d'un CodeBuild projet à utiliser avec Amazon EFS

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

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Choisissez Build projects (Projets de génération) dans le volet de navigation, puis choisissez Create a build project (Créer un projet de build).
3. Dans Project name (Nom de projet), saisissez un nom pour votre projet.
4. Dans Source provider (Fournisseur de source), choisissez le référentiel qui contient l'application Java que vous voulez générer.
5. Entrez les informations, telles que l'URL du référentiel, CodeBuild qui permettent de localiser votre application. Les options sont différentes pour chaque fournisseur de source. Pour de plus amples informations, veuillez consulter [Choose source provider](#).
6. Dans Image de l'environnement, choisissez Image gérée.
7. Dans Operating system (Système d'exploitation), choisissez Amazon Linux 2.
8. Dans Runtime(s) (Exécution(s)), sélectionnez Standard.
9. Dans Image, choisissez aws/codebuild/amazonlinux2-x86_64-standard:4.0.
10. Dans Environment type (Type d'environnement), choisissez Linux.

11. Dans Service role (Rôle de service), choisissez New service role (Nouveau rôle de service). Dans Nom du rôle, entrez le nom du rôle CodeBuild créé pour vous.
12. Développez Additional configuration (Configuration supplémentaire).
13. Sélectionnez Enable this flag if you want to build Docker images or want your builds to get elevated privileges (Activer cet indicateur si vous souhaitez créer des images Docker ou que vos builds reçoivent des privilèges élevés).

 Note

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

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

- (Facultatif) Entrez des options de montage. Si vous laissez le champ Options de montage vide, CodeBuild utilise ses options de montage par défaut :

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

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

18. Pour Build spécification (Spécification de génération), choisissez Insert build commands (Insérer les commandes de génération), puis choisissez Switch to editor (Basculer vers l'éditeur).
19. Entrez les commandes de spécification de construction suivantes dans l'éditeur. Remplacez `<file_system_identifieur>` par l'identificateur que vous avez entré à l'étape 17. Utilisez des lettres majuscules (par exemple, CODEBUILD_MY_EFS).

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

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

CodeBuild et un exemple de résumé d'Amazon EFS

Une fois votre AWS CodeBuild projet créé :

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

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

Résolution des problèmes

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

Rubriques

- [CLIENT_ERROR : échec du montage de « 127.0.0.1 :/». Autorisation refusée](#)
- [CLIENT_ERROR : échec du montage de « 127.0.0.1 :/». Réinitialisation de la connexion par un pair](#)
- [VPC_CLIENT_ERROR : erreur EC2 inattendue : UnauthorizedOperation](#)

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

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

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

CLIENT_ERROR : échec du montage de « 127.0.0.1 :/». Réinitialisation de la connexion par un pair

Cette erreur peut avoir deux causes :

- Le sous-réseau CodeBuild VPC se trouve dans une zone de disponibilité différente de celle de la cible de montage Amazon EFS. Vous pouvez résoudre ce problème en ajoutant un sous-réseau VPC dans la même zone de disponibilité que la cible de montage Amazon EFS.

- Le groupe de sécurité n'est pas autorisé à communiquer avec Amazon EFS. Vous pouvez résoudre ce problème en ajoutant une règle entrante pour autoriser tout le trafic provenant du VPC (ajoutez le bloc CIDR principal pour votre VPC) ou du groupe de sécurité lui-même.

VPC_CLIENT_ERROR : erreur EC2 inattendue : UnauthorizedOperation

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

CodePipeline des échantillons pour CodeBuild

Rubriques

- [AWS CodePipeline intégration avec CodeBuild et constructions par lots](#)
- [AWS CodePipeline intégration avec CodeBuild plusieurs sources d'entrée et échantillons d'artefacts de sortie](#)

AWS CodePipeline intégration avec CodeBuild et constructions par lots

AWS CodeBuild prend désormais en charge les builds par lots. Cet exemple montre comment AWS CodePipeline créer un projet de génération utilisant des builds par lots.

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

Construction par lots avec des artefacts individuels

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

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
```

```
"inputArtifacts": [],
"name": "Source1",
"actionTypeId": {
  "category": "Source",
  "owner": "AWS",
  "version": "1",
  "provider": "S3"
},
"outputArtifacts": [
  {
    "name": "source1"
  }
],
"configuration": {
  "S3Bucket": "<my-input-bucket-name>",
  "S3ObjectKey": "my-source-code-file-name.zip"
},
"runOrder": 1
},
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "<my-other-input-bucket-name>",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
```



```
{
  "inputArtifacts": [
    {
      "name": "source1"
    },
    {
      "name": "source2"
    }
  ],
  "name": "Build",
  "actionTypeId": {
    "category": "Build",
    "owner": "AWS",
    "version": "1",
    "provider": "CodeBuild"
  },
  "outputArtifacts": [
    {
      "name": "build1"
    },
    {
      "name": "build1_artifact1"
    },
    {
      "name": "build1_artifact2"
    },
    {
      "name": "build2_artifact1"
    },
    {
      "name": "build2_artifact2"
    }
  ],
  "configuration": {
    "ProjectName": "my-build-project-name",
    "PrimarySource": "source1",
    "BatchEnabled": "true"
  },
  "runOrder": 1
}
],
"artifactStore": {
```

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

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

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

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

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

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

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

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

Construction par lots avec des artefacts combinés

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

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

```

```
        "S3objectKey": "my-source-code-file-name.zip"
    },
    "runOrder": 1
  },
  {
    "inputArtifacts": [],
    "name": "Source2",
    "actionTypeId": {
      "category": "Source",
      "owner": "AWS",
      "version": "1",
      "provider": "S3"
    },
    "outputArtifacts": [
      {
        "name": "source2"
      }
    ],
    "configuration": {
      "S3Bucket": "<my-other-input-bucket-name>",
      "S3objectKey": "my-other-source-code-file-name.zip"
    },
    "runOrder": 1
  }
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      }
    }
  ]
}
```

```
    },
    "outputArtifacts": [
      {
        "name": "output1 "
      }
    ],
    "configuration": {
      "ProjectName": "my-build-project-name",
      "PrimarySource": "source1",
      "BatchEnabled": "true",
      "CombineArtifacts": "true"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

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

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

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

```
artifacts:
  files:
    - output_file
```

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

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

AWS CodePipeline intégration avec CodeBuild plusieurs sources d'entrée et échantillons d'artefacts de sortie

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

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

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

```
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source1"
    }
  ],
  "configuration": {
    "S3Bucket": "my-input-bucket-name",
    "S3objectKey": "my-source-code-file-name.zip"
  },
  "runOrder": 1
},
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "my-other-input-bucket-name",
    "S3objectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        }
      ],
```

```
    {
      "name": "source2"
    }
  ],
  "name": "Build",
  "actionTypeId": {
    "category": "Build",
    "owner": "AWS",
    "version": "1",
    "provider": "AWS CodeBuild"
  },
  "outputArtifacts": [
    {
      "name": "artifact1"
    },
    {
      "name": "artifact2"
    }
  ],
  "configuration": {
    "ProjectName": "my-build-project-name",
    "PrimarySource": "source1"
  },
  "runOrder": 1
}
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "AWS-CodePipeline-internal-bucket-name"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

Dans ce fichier JSON :

- L'une de vos sources d'entrée doit être désignée comme `PrimarySource`. Cette source est le répertoire dans lequel CodeBuild recherche et exécute votre fichier `buildspec`. Le mot clé

`PrimarySource` est utilisé pour spécifier la source principale dans la configuration section du CodeBuild stage du fichier JSON.

- Chaque source d'entrée est installée dans son propre répertoire. Ce répertoire est stocké dans la variable d'environnement intégrée `$CODEBUILD_SRC_DIR` pour la source principale et `$CODEBUILD_SRC_DIR_YourInputArtifactName` pour toutes les autres sources. Pour le pipeline de cet exemple, les deux répertoires des sources d'entrée sont `$CODEBUILD_SRC_DIR` et `$CODEBUILD_SRC_DIR_source2`. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).
- Les noms des artefacts de sortie spécifiés dans le fichier JSON du pipeline doivent correspondre aux noms des artefacts secondaires définis dans votre fichier buildspec. Ce pipeline utilise le fichier buildspec suivant. Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

```
version: 0.2

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

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

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

Utiliser AWS Config avec un CodeBuild échantillon

AWS Config fournit un inventaire de vos AWS ressources et un historique des modifications de configuration apportées à ces ressources. AWS Config prend désormais en charge en AWS CodeBuild tant que AWS ressource, ce qui signifie que le service peut suivre vos CodeBuild projets. Pour plus d'informations AWS Config, voir [Qu'est-ce que c'est AWS Config ?](#) dans le Guide AWS Config du développeur.

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

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

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

Rubriques

- [Prérequis](#)
- [Configurez AWS Config](#)
- [Rechercher des AWS CodeBuild projets](#)
- [Afficher les détails de AWS CodeBuild configuration dans la AWS Config console](#)

Prérequis

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

Configurez AWS Config

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

Note

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

Rechercher des AWS CodeBuild projets

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

Afficher les détails de AWS CodeBuild configuration dans la AWS Config console

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

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

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

Créer un exemple de notifications pour CodeBuild

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

⚠ Important

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

Exécution de l'exemple

Pour exécuter cet exemple

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

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ]
}
```

```
],  
  "Version": "2012-10-17"  
}
```

Note

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

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

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

Pour créer une rubrique:

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

Topic details: CodeBuildDemoTopic

[Publish to topic](#) [Other topic actions](#)

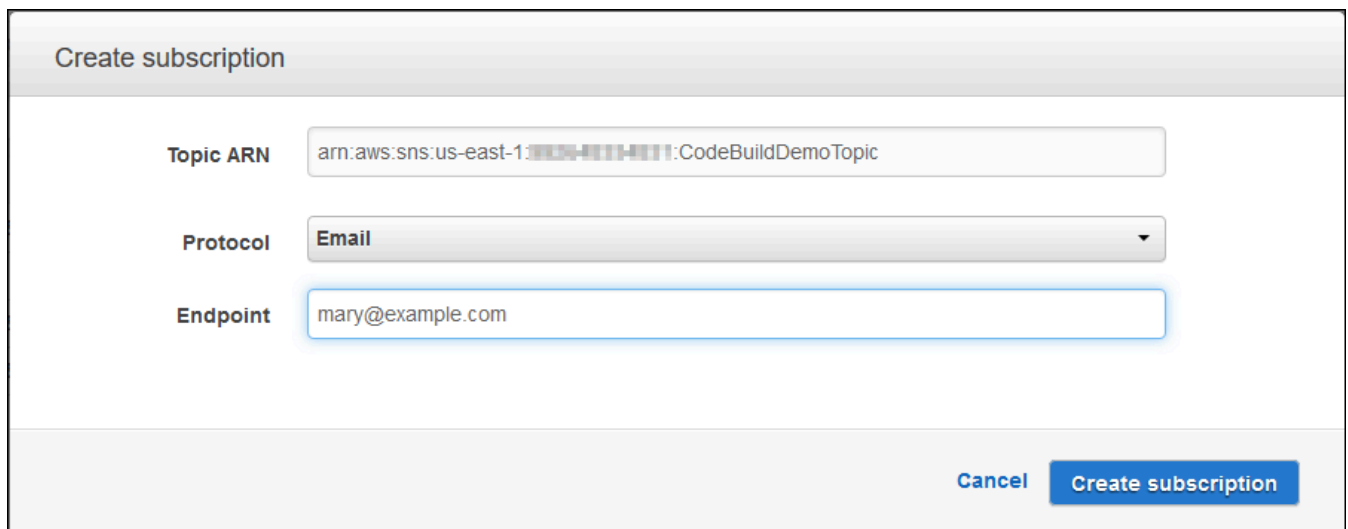
| | |
|--------------|---|
| Topic ARN | arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic |
| Topic owner | 123456789012 |
| Region | us-east-1 |
| Display name | |

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

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

Pour abonner un destinataire à une rubrique :

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



The screenshot shows the 'Create subscription' dialog box. It contains the following fields and values:

- Topic ARN:** arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic
- Protocol:** Email
- Endpoint:** mary@example.com

At the bottom right, there are two buttons: 'Cancel' and 'Create subscription'.

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

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

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

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

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

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

5. Créez une règle dans CloudWatch Events. Pour ce faire, ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch>.
6. Dans le volet de navigation, sous Events, choisissez Rules, puis Create rule.

7. Sur la page Step 1: Create rule page (Étape 1 : Créer une page de règle), les options Event Pattern (Modèle d'événement) et Build event pattern to match events by service (Créer un modèle d'événement pour correspondre aux événements en fonction du service) doivent toujours être activées.
8. Pour Service Name (Nom du service), choisissez CodeBuild. Pour Event Type (Type d'événement), vous devez toujours choisir All Events (Tous les événements).
9. Le code suivant doit être affiché dans l'aperçu du modèle d'événement :

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

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

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

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


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

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

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

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build Phase Change"
  ],
  "detail": {
    "completed-phase": [
      "SUBMITTED",
      "PROVISIONING",
      "DOWNLOAD_SOURCE",
      "INSTALL",
      "PRE_BUILD",
      "BUILD",
      "POST_BUILD",
      "UPLOAD_ARTIFACTS",
      "FINALIZING"
    ],
    "completed-phase-status": [
```


```
"TIMED_OUT",
"STOPPED",
"FAILED",
"SUCCEEDED",
"FAULT",
"CLIENT_ERROR"
],
"project-name": [
  "my-demo-project-1",
  "my-demo-project-2"
]
}
}
```

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

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

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

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

 Note

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ressources connexes

- Pour plus d'informations sur la prise en AWS CodeBuild main, consultez [Mise en route d'AWS CodeBuild à l'aide de la console](#).
- Pour plus d'informations sur la résolution des problèmes dans CodeBuild, voir [Résolution des problèmes AWS CodeBuild](#).
- Pour plus d'informations sur les quotas dans CodeBuild, voir [Quotas pour AWS CodeBuild](#).

Référence du format d'entrée des notifications de génération

CloudWatch fournit des notifications au format JSON.

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

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "build-status": "SUCCEEDED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-artifact.zip"
      },
      "environment": {
        "image": "aws/codebuild/standard:5.0",
        "privileged-mode": false,

```

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

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "UPLOAD_ARTIFACTS",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:26 PM",
```

```

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

```

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

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "completed-phase-context": "[]",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-artifact.zip"
      }
    }
  }
}

```



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

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

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

Créez un échantillon de badges avec CodeBuild


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

Création d'un projet de génération avec des badges de génération activés (console)

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également


inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

4. Dans Source, pour Fournisseur de source, choisissez le type de fournisseur de code source, puis effectuez l'une des actions suivantes :

 Note

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

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

 Important

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

5. Dans Environment (Environnement) :

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

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

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

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

Note

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

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

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

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

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec . yml , target/my-app . jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

9. Développez Additional configuration (Configuration supplémentaire) et choisissez les options appropriées.

10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Création d'un projet de génération avec des badges de génération activés (CLI)

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

Accédez à vos badges AWS CodeBuild de construction

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

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

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

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

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

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

Publiez vos badges CodeBuild de construction

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

```

```

CodeBuild statuts des badges

- **PASSING** : La dernière génération sur la branche indiquée a abouti.
- **FAILING** : La dernière génération sur la branche indiquée a expiré, a échoué, comprend des erreurs ou a été arrêtée.

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

Créez un rapport de test en CodeBuild utilisant l' AWS CLI échantillon

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

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

Rubriques

- [Prérequis](#)
- [Créer un groupe de rapports](#)
- [Configuration d'un projet avec un groupe de rapports](#)
- [Exécution et affichage des résultats d'un rapport](#)

Prérequis

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

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

- Concombre JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- Unit3 XML (.xml)
- TestNG XML (.xml)

- Visual Studio TRX (.trx)

Créez vos cas de test avec n'importe quel framework de test capable de créer des fichiers de rapport dans l'un de ces formats (par exemple, le plug-in Surefire JUnit, TestNG et Cucumber).

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

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

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

Créer un groupe de rapports

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

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

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

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

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

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

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

Configuration d'un projet avec un groupe de rapports

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

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

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

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

Note

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

d'informations, consultez [Créer un rapport de test](#) et [Attribution des noms des groupes de rapports](#).

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

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

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

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

Exécution et affichage des résultats d'un rapport

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

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

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

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

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

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

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

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

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<your-S3-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      }
    }
  ]
}
```

```

    },
    "expired": 1575916770.0,
    "truncated": false,
    "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-
build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",
    "type": "TEST",
    "arn": "<report-arn>",
    "testSummary": {
      "durationInNanoSeconds": 6657770,
      "total": 11,
      "statusCounts": {
        "FAILED": 3,
        "SKIPPED": 7,
        "ERROR": 0,
        "SUCCEEDED": 1,
        "UNKNOWN": 0
      }
    }
  }
],
"reportsNotFound": []
}

```

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

```

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

```

Le résultat se présente comme suit.

```

{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",

```

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

Exemples de Docker pour CodeBuild

Rubriques

- [Docker dans un exemple d'image personnalisée pour CodeBuild](#)
- [Publier une image Docker dans un exemple de référentiel d'images Amazon Elastic Container Registry pour CodeBuild](#)
- [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#)

Docker dans un exemple d'image personnalisée pour CodeBuild

Cet exemple crée et exécute une image Docker en utilisant une image AWS CodeBuild de construction Docker personnalisée (`docker:dind` dans Docker Hub).

Pour savoir comment créer une image Docker en utilisant plutôt une image de construction fournie CodeBuild par le support Docker, consultez notre [Publier une image Docker dans un exemple de référentiel d'images Amazon ECR](#)

Important

L'exécution de cet échantillon peut entraîner des frais sur votre AWS compte. Cela inclut les éventuels frais pour CodeBuild et pour les AWS ressources et les actions liées à Amazon

S3 et à CloudWatch Logs. AWS KMS Pour plus d'informations, consultez [CodeBuild les sections Tarification](#), [Tarification Amazon S3](#), [AWS Key Management Service Tarification](#) et [CloudWatchTarification Amazon](#).

Rubriques

- [Exécution de l'exemple](#)
- [Structure de répertoire](#)
- [Dépôt de](#)
- [Ressources connexes](#)

Exécution de l'exemple

Pour exécuter cet exemple

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

Important

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

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

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

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

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



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

Note

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

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

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

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

Dépôt de

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

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

```
version: 0.2

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

Dockerfile (dans *(root directory name)*)

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Ressources connexes

- Pour plus d'informations sur la prise en AWS CodeBuild main, consultez [Mise en route d'AWS CodeBuild à l'aide de la console](#).
- Pour plus d'informations sur la résolution des problèmes dans CodeBuild, voir [Résolution des problèmes AWS CodeBuild](#).
- Pour plus d'informations sur les quotas dans CodeBuild, voir [Quotas pour AWS CodeBuild](#).

Publier une image Docker dans un exemple de référentiel d'images Amazon Elastic Container Registry pour CodeBuild

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

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

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

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

Important

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

Rubriques

- [Exécution de l'exemple](#)
- [Structure de répertoire](#)
- [Dépôt de](#)
- [Adaptation de l'exemple pour transférer l'image vers Docker Hub](#)
- [Ressources connexes](#)

Exécution de l'exemple

Pour exécuter cet exemple

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

la section [Utilisation des politiques intégrées à l'aide du AWS Management Console](#) guide de l'utilisateur.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

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

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

```
{
```

```
"Statement": [  
  ### BEGIN ADDING STATEMENT HERE ###  
  {  
    "Action": [  
      "ecr:BatchCheckLayerAvailability",  
      "ecr:CompleteLayerUpload",  
      "ecr:GetAuthorizationToken",  
      "ecr:InitiateLayerUpload",  
      "ecr:PutImage",  
      "ecr:UploadLayerPart"  
    ],  
    "Resource": "*",  
    "Effect": "Allow"  
  },  
  ### END ADDING STATEMENT HERE ###  
  ...  
],  
"Version": "2012-10-17"  
}
```

Note

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

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

Important

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

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

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

Si vous utilisez la console pour créer votre projet :

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

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

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

```

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

```

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

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

Structure de répertoire

Cet exemple suppose que cette structure de répertoire est utilisée.

```

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

```

Dépôt de

Cet exemple utilise ces fichiers.

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

```

version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
  build:
    commands:
      - echo Build started on `date`

```

```
- echo Building the Docker image...
- docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
- docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
```

Dockerfile (dans *(root directory name)*)

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

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

Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Adaptation de l'exemple pour transférer l'image vers Docker Hub

Pour transférer l'image Docker vers Docker Hub plutôt que vers Amazon ECR, modifiez le code de cet exemple.

Note

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

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

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

Par ces lignes de code spécifiques à Docker Hub :

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

```
...
```

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

⚠ Important

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

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

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

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

```

Par ces lignes de code :

```
...
```

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

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

Ressources connexes

- Pour plus d'informations sur la prise en main d'AWS CodeBuild, consultez [Mise en route d'AWS CodeBuild à l'aide de la console](#).
- Pour plus d'informations sur la résolution des problèmes dans CodeBuild, voir [Résolution des problèmes AWS CodeBuild](#).
- Pour plus d'informations sur les quotas dans CodeBuild, voir [Quotas pour AWS CodeBuild](#).

Registre privé avec AWS Secrets Manager échantillon pour CodeBuild

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

Note

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

Exigences de l'exemple de registre privé

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

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

Note

Les secrets que vous créez vous seront facturés.

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

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

Création d'un CodeBuild projet avec un registre privé

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

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

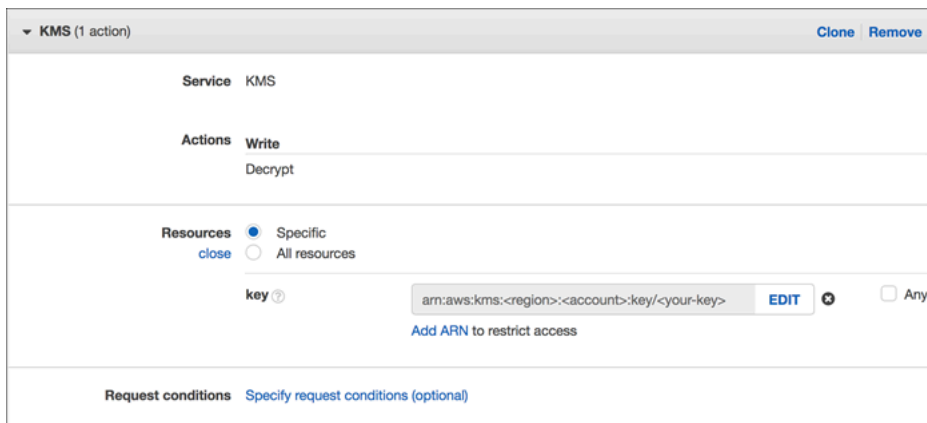
2. Suivez les étapes décrites dans la [section Créer un AWS Secrets Manager secret](#) dans le guide de AWS Secrets Manager l'utilisateur.
 - a. À l'étape 3, dans Choisir le type de secret, choisissez Autre type de secret.
 - b. Dans les paires clé/valeur, créez une paire clé-valeur pour votre nom d'utilisateur Docker Hub et une paire clé-valeur pour votre mot de passe Docker Hub.
 - c. Continuez à suivre les étapes décrites dans [Créer un AWS Secrets Manager secret](#).

- d. À l'étape 5, sur la page Configurer la rotation automatique, désactivez-la car les clés correspondent à vos informations d'identification Docker Hub.
- e. Terminez en suivant les étapes décrites dans [Créer un AWS Secrets Manager secret](#).

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

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

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




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

Note

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

- a. Dans Image d'environnement, sélectionnez Image personnalisée.
- b. Pour Type d'environnement, choisissez Linux ou Windows.
- c. Pour le registre d'images, choisissez Autre registre.

- d. Dans URL de registre externe, entrez l'emplacement de l'image et dans Informations d'identification du registre. Entrez éventuellement l'ARN ou le nom de vos informations d'identification de Secrets Manager.

 Note

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

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

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

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

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

Source Add source

Source 1 - Primary

Source provider
GitHub

Repository
 Public repository Repository in my GitHub account

GitHub repository
 Refresh

Disconnect GitHub account

▼ **Additional configuration**

Git clone depth

Git clone depth - *optional*

Build Status - *optional*
 Report build statuses to source provider when your builds start and finish

Webhook - *optional*
 Rebuild every time a code change is pushed to this repository

Branch filter - *optional*


Enter a regular expression

6. Dans Environment (Environnement) :

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

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.

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

 Note

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

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

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

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

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

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

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

1. Téléchargez vos sources vers un ou plusieurs buckets S3, CodeCommit GitHub, GitHub Enterprise Server ou référentiels Bitbucket.

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

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket>",
      "artifactIdentifier": "artifact1"
    },
    {
      "type": "S3",
      "location": "<other-output-bucket>",
      "artifactIdentifier": "artifact2"
    }
  ],
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  }
}
```

```
  },  
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",  
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"  
}
```

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

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

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

Note

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

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

```
version: 0.2  
  
phases:  
  install:  
    runtime-versions:  
      java: openjdk11  
  build:  
    commands:  
      - cd $CODEBUILD_SRC_DIR_source1  
      - touch file1  
      - cd $CODEBUILD_SRC_DIR_source2  
      - touch file2
```

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

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

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

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

Exemple de projet sans source

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

```
{
```

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

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

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

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

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

Note

Si vous utilisez Docker dans le conteneur de générations, la génération doit s'exécuter en mode privilégié. Pour plus d'informations, consultez [Exécution d'une génération dans AWS CodeBuild](#) et [Création d'un projet de génération dans AWS CodeBuild](#).

Mise à jour de votre version d'environnement d'exécution

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
version: 0.2

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

Dans le fichier `buildspec` :

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

La structure de votre fichier devrait maintenant ressembler à ceci.

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

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

Important

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

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

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

- Pour Environnement :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.
- 7. Choisissez Démarrer la génération.
- 8. Dans Configuration de la génération, acceptez les valeurs par défaut, puis choisissez Démarrer la génération.
- 9. Une fois que la génération est terminée, affichez sa sortie dans l'onglet Journaux de génération. Vous devez voir des résultats similaires à ce qui suit :

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

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

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

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

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

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

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

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

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

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

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

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

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

Spécification de deux environnements d'exécution

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

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

```
package main
import "fmt"

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

```
}
```

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

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

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

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

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

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
      nodejs: 12
```

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

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

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

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

 Important

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

10. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).

11. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#). Laissez tous les paramètres avec leurs valeurs par défaut, sauf pour ces paramètres.
 - Pour Environnement :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez aws/codebuild/amazonlinux2-x86_64-standard:4.0.
12. Choisissez Créer un projet de génération.
13. Choisissez Démarrer la génération.
14. Dans Configuration de la génération, acceptez les valeurs par défaut, puis choisissez Démarrer la génération.
15. Une fois que la génération est terminée, affichez sa sortie dans l'onglet Journaux de génération. Vous devez visualiser des résultats similaires à ce qui suit. Ils montrent la sortie des environnements d'exécution Go et Node.js. Ils montrent également la sortie des applications Go et Node.js.

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

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

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

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time  INSTALL: 0 commands
[Container] Date Time  BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
```

```
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

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

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

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

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

[Container] Date Time Running command npm run test

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

run some tests here
```

Exemple de version source avec AWS CodeBuild

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

- Pour un fournisseur de source Amazon S3, utilisez l'ID de version de l'objet qui représente le fichier ZIP d'entrée de compilation.
- Pour CodeCommit Bitbucket et GitHub Enterprise Server GitHub, utilisez l'une des options suivantes :
 - Demande d'extraction en tant que référence de demande d'extraction (par exemple, `refs/pull/1/head`).
 - Branche en tant que nom de branche.

- ID de validation
- Balise.
- Reference et un ID de validation. La référence peut être l'un des éléments suivants :
 - Une balise (par exemple, `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Une branche (par exemple, `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Une demande d'extraction (par exemple, `refs/pull/1/head^{full-commit-SHA}`).

Note

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

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

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

Note

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

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

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) et [Exécution d'une génération \(console\)](#). Laissez tous les paramètres avec leurs valeurs par défaut, sauf pour les paramètres suivants.
 - Dans Source:
 - Dans le champ Source provider, sélectionnez GitHub. Si vous n'êtes pas connecté à GitHub, suivez les instructions pour vous connecter.
 - Pour Repository (Référentiel), choisissez Public repository (Référentiel public).
 - Pour Repository URL (URL du référentiel), saisissez **https://github.com/aws/aws-sdk-ruby.git**
 - Dans Environment (Environnement) :
 - Pour Image d'environnement, choisissez Managed image (Image gérée).
 - Pour Operating system (Système d'exploitation), choisissez Amazon Linux 2.
 - Pour Runtime(s) (Exécution(s)), sélectionnez Standard.
 - Pour Image, choisissez aws/codebuild/amazonlinux2-x86_64-standard:4.0.
3. Pour Build spécification (Spécifications de génération), choisissez Insérer des commandes de génération, puis choisissez Switch to editor (Passer à l'éditeur).
4. Dans Build commands (Commandes de génération), remplacez le texte de l'espace réservé par ce qui suit :

```
version: 0.2

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

La section `runtime-versions` est obligatoire lorsque vous utilisez l'image standard Ubuntu 2.0. Ici, la version Ruby 2.6 de l'exécution est spécifiée, mais vous pouvez utiliser n'importe

quelle exécution. La commande `echo` affiche la version du code source stocké dans la variable d'environnement `CODEBUILD_RESOLVED_SOURCE_VERSION`.

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

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

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

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

Ces étapes vous montrent comment créer une version de génération à l'aide de la même version de la source. Cette fois, la version de la source est spécifiée à l'aide d'une référence avec l'ID de validation.

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

1. Dans le volet de navigation de gauche, choisissez Projets de génération, puis choisissez le projet que vous avez créé précédemment.
2. Choisissez Démarrer la génération.
3. Dans Version de la source, saisissez **`refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}`**. Il s'agit du même ID de validation et d'une référence à une branche dans le format `refs/heads/branchname^{full-commit-SHA}`.
4. Choisissez Démarrer la génération.

5. Une fois la génération terminée, vous devez voir ce qui suit :

- Sur l'onglet Journaux de génération, la version de la source du projet qui a été utilisée. Voici un exemple.

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

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

- Sur l'onglet Variables d'environnement, Resolved source version (Version de source résolue) correspond à l'ID de validation utilisé pour créer la validation.
- Sur l'onglet Détails de Détail de la phase, la durée de la phase DOWNLOAD_SOURCE doit être inférieure à la durée lorsque vous avez utilisé uniquement l'ID de validation pour spécifier la version de votre code source.

Exemples de référentiels sources tiers pour CodeBuild

Rubriques

- [Exemple de pull request et de filtre Webhook de Bitbucket pour CodeBuild](#)
- [GitHub Exemple de serveur d'entreprise pour CodeBuild](#)
- [GitHub exemple de pull request et de filtre Webhook pour CodeBuild](#)

Exemple de pull request et de filtre Webhook de Bitbucket pour CodeBuild

AWS CodeBuild prend en charge les webhooks lorsque le référentiel source est Bitbucket. Cela signifie que pour un projet de CodeBuild compilation dont le code source est stocké dans un dépôt Bitbucket, les webhooks peuvent être utilisés pour reconstruire le code source chaque fois qu'une modification de code est envoyée au référentiel. Pour de plus amples informations, veuillez consulter [Événements du webhook Bitbucket](#).

Cet exemple vous montre comment créer une demande d'extraction à l'aide d'un référentiel Bitbucket. Il vous montre également comment utiliser un webhook Bitbucket pour déclencher la création CodeBuild d'une version d'un projet.

Note

Lors de l'utilisation de webhooks, il est possible pour un utilisateur de déclencher une compilation inattendue. Pour atténuer ce risque, voir [Bonnes pratiques d'utilisation des webhooks](#).

Rubriques

- [Prérequis](#)
- [Création d'un projet de génération avec Bitbucket comme référentiel source et activation des webhooks](#)
- [Déclenchement d'une génération avec un webhook Bitbucket](#)

Prérequis

Pour exécuter cet exemple, vous devez connecter votre AWS CodeBuild projet à votre compte Bitbucket.

Note

CodeBuild a mis à jour ses autorisations avec Bitbucket. Si vous avez précédemment connecté votre projet à Bitbucket et que vous recevez maintenant une erreur de connexion Bitbucket, vous devez vous reconnecter pour CodeBuild autoriser la gestion de vos webhooks.

Création d'un projet de génération avec Bitbucket comme référentiel source et activation des webhooks

Les étapes suivantes décrivent comment créer un AWS CodeBuild projet avec Bitbucket comme référentiel source et comment activer les webhooks.

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

3. Choisissez Créer un projet de génération.
4. Dans Configuration de projet :

Nom du projet

Entrez un nom pour ce projet de construction. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

5. Dans Source:

Fournisseur de source

Choisissez Bitbucket. Suivez les instructions pour vous connecter (ou vous reconnecter) à Bitbucket, puis choisissez Autoriser.

Référentiel

Choisissez Repository dans mon compte Bitbucket.

Si vous n'êtes pas encore connecté à votre compte Bitbucket, entrez votre nom d'utilisateur Bitbucket et le mot de passe de l'application, puis sélectionnez Enregistrer les informations d'identification Bitbucket.

Référentiel Bitbucket

Entrez l'URL de votre dépôt Bitbucket.

6. Dans Événements webhook de la source principale, sélectionnez ce qui suit.

Note

La section des événements du webhook de la source principale n'est visible que si vous avez choisi Repository in my Bitbucket account à l'étape précédente.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.

3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations sur les types d'événements et les filtres Bitbucket Webhook, consultez.

[Événements du webhook Bitbucket](#)

7. Dans Environment (Environnement) :

Image de l'environnement

Sélectionnez l'une des méthodes suivantes :

Pour utiliser une image Docker gérée par AWS CodeBuild :

Choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.

Pour utiliser une autre image Docker :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS

Pour utiliser une image Docker privée :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur.

Rôle de service

Sélectionnez l'une des méthodes suivantes :

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

Note

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

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

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

9. Dans Artefacts:

Type

Sélectionnez l'une des méthodes suivantes :

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

d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.

- Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
- Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, `appspect.yml`, `target/my-app.jar`). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

Configuration supplémentaire

Développez Additional configuration (Configuration supplémentaire) et définissez les options dont vous avez besoin.

10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Déclenchement d'une génération avec un webhook Bitbucket

Pour un projet qui utilise des webhooks Bitbucket, AWS CodeBuild crée une version lorsque le référentiel Bitbucket détecte une modification dans votre code source.

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération, puis sélectionnez un projet associés à un référentiel Bitbucket avec webhooks. Pour plus d'informations sur la création d'un projet Webhook Bitbucket, consultez [the section called "Création d'un projet de génération avec Bitbucket comme référentiel source et activation des webhooks"](#)
3. Modifiez le code du référentiel Bitbucket de votre projet.
4. Créez une demande d'extraction sur votre référentiel Bitbucket. Pour de plus amples informations, veuillez consulter des informations sur la [création d'une demande d'extraction](#).
5. Dans la page des webhooks Bitbucket, choisissez View request (Afficher la demande) pour afficher la liste des événements récents.
6. Choisissez Afficher les détails pour voir les détails de la réponse renvoyée par CodeBuild. Elle peut ressembler à ceci :

```
"response":"Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
"statusCode":200
```

7. Accédez à la page de demande d'extraction Bitbucket pour voir le statut de la génération.

GitHub Exemple de serveur d'entreprise pour CodeBuild

AWS CodeBuild prend en charge GitHub Enterprise Server en tant que référentiel source. Cet exemple montre comment configurer vos CodeBuild projets lorsqu'un certificat est installé dans votre référentiel GitHub Enterprise Server. Il montre également comment activer les webhooks afin de CodeBuild recompiler le code source chaque fois qu'une modification de code est transmise à votre référentiel GitHub Enterprise Server.

Prérequis

1. Générez un jeton d'accès personnel pour votre CodeBuild projet. Nous vous recommandons de créer un utilisateur GitHub Enterprise et de générer un jeton d'accès personnel pour cet utilisateur. Copiez-le dans votre presse-papiers afin de pouvoir l'utiliser lors de la création de votre CodeBuild projet. Pour plus d'informations, consultez la section [Création d'un jeton d'accès personnel pour la ligne de commande](#) sur le site Web d' GitHub aide.

Lorsque vous créez le jeton d'accès personnel, incluez la portée du référentiel dans la définition.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

| | | |
|-------------------------------------|-----------------|--------------------------------------|
| <input checked="" type="checkbox"/> | repo | Full control of private repositories |
| <input checked="" type="checkbox"/> | repo:status | Access commit status |
| <input checked="" type="checkbox"/> | repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> | public_repo | Access public repositories |

2. Téléchargez votre certificat depuis GitHub Enterprise Server. CodeBuild utilise le certificat pour établir une connexion SSL fiable avec le référentiel.

Clients Linux/macOS :

À partir d'une fenêtre de terminal , exécutez la commande suivante :

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```


Remplacez les espaces réservés de la commande par les valeurs suivantes :

HOST. Adresse IP de votre référentiel GitHub Enterprise Server.

PORTNUMBER. Numéro de port que vous utilisez pour vous connecter (par exemple, 443).

folder. Dossier dans lequel vous avez téléchargé votre certificat.


filename. Nom du fichier de votre certificat.

 Important

Enregistrez le certificat en tant que fichier .pem.


Clients Windows:

Utilisez votre navigateur pour télécharger votre certificat depuis GitHub Enterprise Server. Pour afficher les détails de certificat du site, choisissez l'icône en forme de cadenas. Pour plus d'informations sur la manière d'exporter le certificat, consultez la documentation de votre navigateur.

 Important

Enregistrez le certificat en tant que fichier .pem.

3. Chargez votre fichier de certificat dans un compartiment S3. Pour de plus amples informations sur la façon de créer un compartiment S3, veuillez consulter [Comment créer un compartiment S3 ?](#) Pour de plus amples informations sur le chargement d'objets dans un compartiment S3, veuillez consulter [Comment charger des fichiers ou dossiers vers un compartiment ?](#)

 Note

Ce compartiment doit se trouver dans la même AWS région que vos builds. Par exemple, si vous demandez d' CodeBuild exécuter un build dans la région USA Est (Ohio), le bucket doit se trouver dans la région USA Est (Ohio).

Créez un projet de build avec GitHub Enterprise Server comme référentiel source et activez les webhooks (console)

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, dans Source provider, choisissez GitHub Enterprise.
 - Pour Personal Access Token, collez le jeton que vous avez copié dans le Presse-papiers et choisissez Save Token. Dans URL du référentiel, entrez l'URL de votre référentiel GitHub Enterprise Server.

Note

Vous n'avez besoin de saisir et d'enregistrer le jeton d'accès personnel qu'une seule fois. Tous les futurs AWS CodeBuild projets utilisent ce jeton.

- Dans URL du référentiel, saisissez le chemin d'accès à votre référentiel, y compris le nom du référentiel.
- Développez Additional configuration (Configuration supplémentaire).
- Sélectionnez Rebuild every time a code change is pushed to this repository (Regénérer chaque fois qu'une modification de code est transmise à ce référentiel) pour régénérer chaque fois qu'une modification de code est transmise à ce référentiel.
- Sélectionnez Activer le protocole SSL non sécurisé pour ignorer les avertissements SSL lorsque vous vous connectez au référentiel de votre projet GitHub Enterprise Server.

Note

Nous vous recommandons d'utiliser Insecure SSL (SSL non sécurisé) uniquement pour les tests. Cette option ne doit pas être utilisée dans un environnement de production.

Source Add source

Source 1 - Primary

Source provider

GitHub Enterprise

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ Additional configuration
Git clone depth, Insecure SSL

Git clone depth - *optional*

1

Webhook - *optional*

Rebuild every time a code change is pushed to this repository

Branch filter - *optional*

Enter a regular expression

Insecure SSL - *optional*
Enable this flag to ignore SSL warnings while connecting to project source.

Enable insecure SSL

5. Dans Environment (Environnement) :

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

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other

registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS

- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

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

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

Note

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

7. Développez Additional configuration (Configuration supplémentaire).

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise. CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent. CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources des VPC.

Pour de plus amples informations, veuillez consulter [Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud](#).

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

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

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

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

10. Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :


- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de

répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

 Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

 Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Mise en cache de génération dans AWS CodeBuild](#).

11. Choisissez Créer un projet de génération. Sur la page de projet de génération, choisissez Démarrer la génération.
12. Si vous avez activé des webhooks dans Source, une boîte de dialogue Create webhook (Créer un webhook) s'affiche avec des valeurs pour Payload URL (URL de charge utile) et Secret.

 Important

La boîte de dialogue Créer un webhook n'apparaît qu'une seule fois. Copiez l'URL et la clé secrète de la charge utile. Vous en avez besoin lorsque vous ajoutez un webhook dans GitHub Enterprise Server.

Si vous devez générer à nouveau une URL de charge utile et une clé secrète, vous devez d'abord supprimer le webhook de votre référentiel GitHub Enterprise Server. Dans votre projet CodeBuild, désélectionnez la case à cocher Webhook, puis choisissez Enregistrer. Vous pouvez ensuite créer ou mettre à jour un projet CodeBuild dont la case Webhook est cochée. La boîte de dialogue Créer un webhook apparaît à nouveau.

13. Dans GitHub Enterprise Server, choisissez le référentiel dans lequel votre CodeBuild projet est stocké.
14. Choisissez Settings (Paramètres), Hooks & services, puis Add webhook (Ajouter un webhook).
15. Entrez l'URL et la clé secrète de la charge utile, acceptez les valeurs par défaut pour les autres champs, puis choisissez Add webhook.

The screenshot shows the 'Add webhook' configuration page in GitHub Enterprise Server. At the top, there is a navigation bar with tabs for 'requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Settings' tab is active. Below the navigation bar, the page title is 'Webhooks / Add webhook'. A descriptive paragraph states: 'We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.' The form contains the following fields: 'Payload URL *' (a text input field), 'Content type' (a dropdown menu set to 'application/json'), and 'Secret' (a password input field). Below these fields, there is a checkbox for 'By default, we verify SSL certificates when delivering payloads.' with a 'Disable SSL verification' button. The 'Which events would you like to trigger this webhook?' section has three radio button options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'. At the bottom, there is a checked checkbox for 'Active' with the subtext 'We will deliver event details when this hook is triggered.' and a green 'Add webhook' button.

16. Retournez à votre CodeBuild projet. Fermez la boîte de dialogue Créer un webhook et choisissez Démarrer la génération.

GitHub exemple de pull request et de filtre Webhook pour CodeBuild

AWS CodeBuild prend en charge les webhooks lorsque le référentiel source l'est GitHub. Cela signifie que pour un projet de CodeBuild compilation dont le code source est stocké dans un GitHub référentiel, les webhooks peuvent être utilisés pour reconstruire le code source chaque fois qu'une modification de code est envoyée au référentiel. Pour CodeBuild des exemples, voir [AWS CodeBuild Échantillons](#).

Note

Lors de l'utilisation de webhooks, il est possible pour un utilisateur de déclencher une compilation inattendue. Pour atténuer ce risque, voir [Bonnes pratiques d'utilisation des webhooks](#).

Créez un projet de build avec GitHub comme référentiel source et activez les webhooks (console)

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Choisissez Créer un projet de génération.
4. Dans Configuration de projet :

Nom du projet

Entrez un nom pour ce projet de construction. Les noms des projets de construction doivent être uniques pour chaque AWS compte. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

5. Dans Source:

Fournisseur de source

Choisissez GitHub. Suivez les instructions pour vous connecter (ou vous reconnecter) à, GitHub puis choisissez Autoriser.

Référentiel

Choisissez Repository dans mon GitHub compte.

GitHub référentiel

Entrez l'URL de votre GitHub dépôt.

6. Dans Événements webhook de la source principale, sélectionnez ce qui suit.

Note

La section des événements du webhook de la source principale n'est visible que si vous avez choisi Repository dans mon GitHub compte à l'étape précédente.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations sur les types d'événements et les filtres GitHub Webhook, consultez [GitHub événements webhook](#).

7. Dans Environment (Environnement) :

Image de l'environnement

Sélectionnez l'une des méthodes suivantes :

Pour utiliser une image Docker gérée par AWS CodeBuild :

Choisissez Image gérée, puis sélectionnez Système d'exploitation, Runtime (s), Image et Version de l'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.

Pour utiliser une autre image Docker :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS

Pour utiliser une image Docker privée :

Choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur.

Rôle de service

Sélectionnez l'une des méthodes suivantes :

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

Note

Lorsque vous utilisez la console pour créer ou mettre à jour un projet de génération, vous pouvez créer un rôle de CodeBuild service en même temps. Par défaut, le rôle fonctionne avec ce projet de génération uniquement. Si vous utilisez la console pour

associer ce rôle de service à un autre projet de génération, le rôle est mis à jour pour fonctionner avec l'autre projet de génération. Un rôle de service peut fonctionner avec 10 projets de génération maximum.

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

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

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

9. Dans Artefacts:

Type

Sélectionnez l'une des méthodes suivantes :

- Si vous ne souhaitez pas créer d'artefacts de sortie de génération, choisissez Aucun artefact.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. Par défaut, le nom d'un artefact est le nom du projet. Si vous souhaitez utiliser un autre nom, entrez-le dans la zone Nom des artefacts. Si vous souhaitez générer un fichier ZIP, incluez l'extension zip.
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec.yml, target/my-app.jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

Configuration supplémentaire

Développez Additional configuration (Configuration supplémentaire) et définissez les options dont vous avez besoin.

10. Choisissez Créer un projet de génération. Sur la page Révision, choisissez Démarrer la génération pour exécuter la génération.

Contrôles de vérification

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :
 - Cliquez sur le lien du projet de build avec des webhooks que vous souhaitez réviser, puis choisissez Build details (Détails de build).
 - Cliquez sur le bouton à côté du projet de génération avec les webhooks que vous souhaitez vérifier, choisissez Afficher les détails, puis choisissez l'onglet Détails de la construction.
4. Dans Événements du webhook de la source principale, choisissez le lien URL du webhook.
5. Dans votre GitHub dépôt, sur la page Paramètres, sous Webhooks, vérifiez que les Pull Requests et Push sont sélectionnés.
6. Dans les paramètres GitHub de votre profil, sous Paramètres personnels, Applications, Applications OAuth autorisées, vous devriez voir que votre application a été autorisée à accéder à la AWS région que vous avez sélectionnée.

Exemple d'utilisation de la gestion sémantique des versions pour nommer des artefacts de génération

Cet exemple contient des exemples de fichiers buildspec qui montrent comment spécifier le nom d'un artefact qui est créé au moment de la génération. Un nom spécifié dans un fichier buildspec peut comprendre des commandes Shell et des variables d'environnement afin de le rendre unique. Un nom que vous spécifiez dans un fichier buildspec se substitue au nom que vous saisissez dans la console lorsque vous créez votre projet.

Si vous procédez à plusieurs générations, le nom d'artefact spécifié dans le fichier buildspec permet de vous assurer que les noms de vos fichiers d'artefact de sortie sont uniques. Par exemple, vous pouvez utiliser une date et un horodatage inséré dans le nom d'un artefact au moment de la génération.

Si vous souhaitez remplacer le nom de l'artefact que vous avez saisi dans la console par un nom dans le fichier buildspec, procédez comme suit :

1. Configurez votre projet de génération afin de remplacer le nom de l'artefact par un nom du fichier buildspec.
 - Si vous utilisez la console pour créer votre projet de génération, sélectionnez Activer la gestion sémantique des versions. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).
 - Si vous utilisez le AWS CLI, définissez le sur `overrideArtifactName true` dans le fichier au format JSON transmis à `create-project` Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).
 - Si vous utilisez l' AWS CodeBuild API, placez l'`overrideArtifactName`indicateur sur l'`ProjectArtifact`objet lorsqu'un projet est créé ou mis à jour ou qu'une construction est lancée.
2. Spécifiez un nom dans le fichier buildspec. Utilisez l'exemple de fichier buildspec suivant comme guide.

Cet exemple Linux vous montre comment spécifier un nom d'artefact qui comprend la date de création de la génération :

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
name: myname-${date +%Y-%m-%d}
```

Cet exemple Linux vous montre comment spécifier un nom d'artefact qui utilise une variable d' CodeBuild environnement. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
name: myname-$AWS_REGION
```

Cet exemple Windows vous montre comment spécifier un nom d'artefact qui comprend la date et l'heure de création de la génération :

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Cet exemple Windows montre comment spécifier un nom d'artefact qui utilise une variable déclarée dans le fichier buildspec et une variable d'environnement. CodeBuild Pour plus d'informations, consultez [Variables d'environnement dans les environnements de génération](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
```

```
build:
  commands:
    - cd samples/helloworld
    - dotnet restore
    - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Pour plus d'informations, consultez [Référence de spécification de construction pour CodeBuild](#).

Exemples Microsoft Windows pour CodeBuild

Ces exemples utilisent un environnement de AWS CodeBuild génération exécutant Microsoft Windows Server 2019, le .NET Framework et le SDK .NET Core pour créer des fichiers d'exécution à partir de code écrit en F# et Visual Basic.

Important

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

Exécution des exemples

Pour exécuter ces exemples :

1. Créez les fichiers comme décrit dans les sections « Structure du répertoire » et « Fichiers » de cette rubrique, puis chargez-les dans un compartiment d'entrée S3 CodeCommit ou un GitHub référentiel or.

Important

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

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

2. Créez un projet de génération. Le projet de génération doit utiliser `mcr.microsoft.com/dotnet/framework/sdk:4.8` image pour créer des projets .NET Framework.

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

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Exécutez le build et suivez les étapes décrites dans [Exécution d'une génération](#).
4. Pour obtenir l'artefact de sortie de génération, dans le compartiment de sortie S3, téléchargez le fichier *windows-build-output-artifact.zip* sur votre ordinateur ou instance local. Extrayez le contenu pour accéder au moteur d'exécution et aux autres fichiers.
 - Le fichier d'exécution de l'exemple F# utilisant le .NET Framework se trouve dans le `FSharpHelloWorld\bin\Debug` répertoire. `FSharpHelloWorld.exe`

- Le fichier d'exécution de l'exemple Visual Basic utilisant le .NET Framework se trouve dans le VBHelloWorld\bin\Debug répertoire. VBHelloWorld.exe

Structure de répertoire

Ces exemples supposent les structures de répertoire suivantes.

F# et .NET Framework

```
(root directory name)
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
### App.config
### AssemblyInfo.fs
### FSharpHelloWorld.fsproj
### Program.fs
```

Visual Basic et .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
### App.config
### HelloWorld.vb
### VBHelloWorld.vbproj
### My Project
### Application.Designer.vb
### Application.myapp
### AssemblyInfo.vb
### Resources.Designer.vb
### Resources.resx
### Settings.Designer.vb
### Settings.settings
```

Dépôt de

Ces exemples utilisent les fichiers suivants.

F# et .NET Framework

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

```
version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln (dans *(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
```

```
HideSolutionNode = FALSE
EndGlobalSection
EndGlobal
```

App.config (dans *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

AssemblyInfo.fs (dans *(root directory name)*\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
```

```
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]

do
  ()
```

FSharpHelloWorld.fsproj (dans *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <Tailcalls>>false</Tailcalls>
    <OutputPath>bin\Debug\<</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
```

```

    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <Tailcalls>true</Tailcalls>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="mscorlib" />
    <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
      <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Numerics" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="AssemblyInfo.fs" />
    <Compile Include="Program.fs" />
    <None Include="App.config" />
  </ItemGroup>
  <PropertyGroup>
    <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''">11</
MinimumVisualStudioVersion>
  </PropertyGroup>
  <Choose>
    <When Condition="'$(VisualStudioVersion)' == '11.0'">
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </When>
    <Otherwise>

```

```

    <PropertyGroup Condition="Exists('${MSBuildExtensionsPath32}\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
    </PropertyGroup>
  </Otherwise>
</Choose>
<Import Project="$(FSharpTargetsPath)" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Program.fs (dans *(root directory name)*\FSharpHelloWorld):

```

// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code

```

Visual Basic et .NET Framework

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

```

version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:

```

```

- '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
PackagesDirectory $env:PACKAGE_DIRECTORY'
- '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\VBHelloWorld\bin\Debug\*

```

VBHelloWorld.sln (dans *(root directory name)*):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld
\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal

```

App.config (dans *(root directory name)\VBHelloWorld*):

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>

```

HelloWorld.vb (dans *(root directory name)*\VBHelloWorld):

```
Module HelloWorld

    Sub Main()
        MsgBox("Hello World")
    End Sub

End Module
```

VBHelloWorld.vbproj (dans *(root directory name)*\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\<</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
```



```
<PlatformTarget>AnyCPU</PlatformTarget>
<DebugType>pdbonly</DebugType>
<DefineDebug>>false</DefineDebug>
<DefineTrace>>true</DefineTrace>
<Optimize>>true</Optimize>
<OutputPath>bin\Release\  
</OutputPath>
<DocumentationFile>VBHelloWorld.xml</DocumentationFile>
<NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
</PropertyGroup>
<PropertyGroup>
  <OptionExplicit>On</OptionExplicit>
</PropertyGroup>
<PropertyGroup>
  <OptionCompare>Binary</OptionCompare>
</PropertyGroup>
<PropertyGroup>
  <OptionStrict>Off</OptionStrict>
</PropertyGroup>
<PropertyGroup>
  <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Xml" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
  <Import Include="Microsoft.VisualBasic" />
  <Import Include="System" />
  <Import Include="System.Collections" />
  <Import Include="System.Collections.Generic" />
  <Import Include="System.Data" />
  <Import Include="System.Diagnostics" />
  <Import Include="System.Linq" />
  <Import Include="System.Xml.Linq" />
  <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
  <Compile Include="HelloWorld.vb" />
</ItemGroup>
```

```
<Compile Include="My Project\AssemblyInfo.vb" />
<Compile Include="My Project\Application.Designer.vb">
  <AutoGen>True</AutoGen>
  <DependentUpon>Application.myapp</DependentUpon>
</Compile>
<Compile Include="My Project\Resources.Designer.vb">
  <AutoGen>True</AutoGen>
  <DesignTime>True</DesignTime>
  <DependentUpon>Resources.resx</DependentUpon>
</Compile>
<Compile Include="My Project\Settings.Designer.vb">
  <AutoGen>True</AutoGen>
  <DependentUpon>Settings.settings</DependentUpon>
  <DesignTimeSharedInput>True</DesignTimeSharedInput>
</Compile>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="My Project\Resources.resx">
    <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.vb</LastGenOutput>
    <CustomToolNamespace>My.Resources</CustomToolNamespace>
    <SubType>Designer</SubType>
  </EmbeddedResource>
</ItemGroup>
<ItemGroup>
  <None Include="My Project\Application.myapp">
    <Generator>MyApplicationCodeGenerator</Generator>
    <LastGenOutput>Application.Designer.vb</LastGenOutput>
  </None>
  <None Include="My Project\Settings.settings">
    <Generator>SettingsSingleFileGenerator</Generator>
    <CustomToolNamespace>My</CustomToolNamespace>
    <LastGenOutput>Settings.Designer.vb</LastGenOutput>
  </None>
  <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
```

```
-->
</Project>
```

Application.Designer.vb (dans *(root directory name)*\VBHelloWorld\My Project):

```
'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On
```

Application.myapp (dans *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb (dans *(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.
```

```
' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>

'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

Resources.Designer.vb (dans *(root directory name)*\VBHelloWorld\My Project):

```
'-----
' <auto-generated>
' This code was generated by a tool.
' Runtime Version:4.0.30319.42000
'
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On
```

Namespace My.Resources

```

'This class was auto-generated by the StronglyTypedResourceBuilder
'class via a tool like ResGen or Visual Studio.
'To add or remove a member, edit your .ResX file then rerun ResGen
'with the /str option, or rebuild your VS project.
'''<summary>
''' A strongly-typed resource class, for looking up localized strings, etc.
'''</summary>

<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
Friend Module Resources

    Private resourceMan As Global.System.Resources.ResourceManager

    Private resourceCulture As Global.System.Globalization.CultureInfo

    '''<summary>
    ''' Returns the cached ResourceManager instance used by this class.
    '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
    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.EditorBrowsable
-
    Friend Property Culture() As Global.System.Globalization.CultureInfo
        Get
            Return resourceCulture
        End Get
        Set(ByVal value As Global.System.Globalization.CultureInfo)
            resourceCulture = value
        End Set
    End Property
End Module
End Namespace

```

Resources.resx (dans (*root directory name*)\VBHelloWorld\My Project):

```

<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema

```

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

```

... ado.net/XML headers & schema ...
<resheader name="resmimetype">text/microsoft-resx</resheader>
<resheader name="version">2.0</resheader>
<resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
<resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
<data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
  <value>[base64 mime encoded serialized .NET Framework object]</value>
</data>

```

```

<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
  <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
  <comment>This is a comment</comment>
</data>

```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

```

mimetype: application/x-microsoft.net.object.binary.base64
value   : The object must be serialized with
         : System.Serialization.Formatters.Binary.BinaryFormatter
         : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.soap.base64
value   : The object must be serialized with
         : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
         : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.bytearray.base64
value   : The object must be serialized into a byte array
         : using a System.ComponentModel.TypeConverter
         : and then encoded with base64 encoding.

```

```
-->
```

```

<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">

```

```

<xsd:element name="metadata">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" />
    <xsd:attribute name="type" type="xsd:string" />
    <xsd:attribute name="mimetype" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="assembly">
  <xsd:complexType>
    <xsd:attribute name="alias" type="xsd:string" />
    <xsd:attribute name="name" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="data">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="resheader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
  <value>text/microsoft-resx</value>

```



```

</resheader>
<resheader name="version">
  <value>2.0</value>
</resheader>
<resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
</root>

```

Settings.Designer.vb (dans *(root directory name)\VBHelloWorld\My Project*):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

Option Strict On
Option Explicit On

Namespace My

  <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
"11.0.0.0"), _
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
  Partial Friend NotInheritable Class MySettings
    Inherits Global.System.Configuration.ApplicationSettingsBase

```

```

Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
MySettings), MySettings)

#Region "My.Settings Auto-Save Functionality"
  #If _MyType = "WindowsForms" Then
    Private Shared addedHandler As Boolean

    Private Shared addedHandlerLockObject As New Object

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
    Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
      If My.Application.SaveMySettingsOnExit Then
        My.Settings.Save()
      End If
    End Sub
  #End If
#End Region

Public Shared ReadOnly Property [Default]() As MySettings
  Get

    #If _MyType = "WindowsForms" Then
      If Not addedHandler Then
        SyncLock addedHandlerLockObject
          If Not addedHandler Then
            AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
            addedHandler = True
          End If
        End SyncLock
      End If
    #End If
    Return defaultInstance
  End Get
End Property
End Class
End Namespace

Namespace My

  <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _

```

```
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute())> _
Friend Module MySettingsProperty

    <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
    Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
        Get
            Return Global.VBHelloWorld.My.MySettings.Default
        End Get
    End Property
End Module
End Namespace
```

Settings.settings (dans *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

Planification d'une génération dans AWS CodeBuild

Avant d'utiliser AWS CodeBuild, vous devez répondre aux questions suivantes :

1. Où est stocké le code source ? CodeBuild prend actuellement en charge la création à partir des fournisseurs de référentiels de code source suivants. Le code source doit contenir un fichier de spécification de génération (buildspec). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Vous pouvez déclarer une spécification de construction dans la définition d'un projet de construction.

| Fournisseur du référentiel | Obligatoire | Documentation |
|----------------------------|--|--|
| CodeCommit | Nom du référentiel. (Facultatif) ID de validation associé au code source. | Consultez ces rubriques dans le Guide de l'utilisateur AWS CodeCommit : Création d'un CodeCommit référentiel Créez un commit dans CodeCommit |
| Amazon S3 | Nom du compartiment d'entrée. Nom d'objet correspondant au fichier ZIP d'entrée de génération qui contient le code source. (Facultatif) ID de version associé au fichier ZIP | Consultez les rubriques suivantes dans le guide de démarrage d'Amazon S3 : Créez un compartiment Ajout d'un objet dans un compartiment |

| Fournisseur du référentiel | Obligatoire | Documentation |
|----------------------------|--|--|
| | d'entrée de génération. | |
| GitHub | Nom du référentiel. (Facultatif) ID de validation associé au code source. | Consultez cette rubrique sur le site Web GitHub d'aide : Create a repo |
| Bitbucket | Nom du référentiel. (Facultatif) ID de validation associé au code source. | Consultez cette rubrique sur le site web de la documentation Bitbucket Cloud : "Création d'un référentiel |

2. Quelles commandes de génération devez-vous exécuter et dans quel ordre ? Par défaut, CodeBuild télécharge l'entrée de build depuis le fournisseur que vous spécifiez et télécharge la sortie de build dans le bucket que vous spécifiez. Vous utilisez la spécification de génération pour indiquer comment transformer l'entrée de génération téléchargée en la sortie de génération attendue. Pour plus d'informations, consultez le [Référence des spécifications de génération](#).
3. Quels sont les environnements d'exécution et les outils dont vous avez besoin pour exécuter la génération ? Par exemple, effectuez-vous une génération pour Java, Ruby, Python ou Node.js ? La génération a-t-elle besoin de Maven ou Ant, ou d'un compilateur pour Java, Ruby ou Python ? La génération a-t-elle besoin de Git, de l'AWS CLI, ou d'autres outils ?

CodeBuild exécute des builds dans des environnements de build qui utilisent des images Docker. Ces images Docker doivent être stockées dans un type de référentiel pris en charge par CodeBuild. Il s'agit notamment du référentiel d'images CodeBuild Docker, de Docker Hub et d'Amazon Elastic Container Registry (Amazon ECR). Pour plus d'informations sur le référentiel d'images CodeBuild Docker, consultez [Images Docker fournies par CodeBuild](#).

4. Avez-vous besoin de AWS ressources qui ne sont pas fournies automatiquement par CodeBuild ? Dans l'affirmative, de quelles politiques de sécurité ces ressources ont-elles besoin ? Par exemple, vous devrez peut-être modifier le rôle CodeBuild de service CodeBuild pour autoriser l'utilisation de ces ressources.
5. Voulez-vous CodeBuild travailler avec votre VPC ? Si tel est le cas, vous avez besoin de l'ID de VPC, des ID de sous-réseau et des ID de groupe de sécurité pour votre configuration de VPC. Pour plus d'informations, consultez [Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Une fois que vous avez répondu à ces questions, vous devriez disposer des paramètres et des ressources dont vous avez besoin pour exécuter une génération avec succès. Pour exécuter votre génération, vous pouvez :

- Utilisez la console AWS CodeBuild, l'AWS CLI ou les kits SDK AWS. Pour plus d'informations, consultez [Exécution d'une génération](#).
- Créez ou identifiez un pipeline dans AWS CodePipeline, puis ajoutez une action de génération ou de test demandant de CodeBuild tester automatiquement votre code, d'exécuter votre build, ou les deux. Pour plus d'informations, consultez [À utiliser CodePipeline avec CodeBuild](#).

Référence de spécification de construction pour CodeBuild

Cette rubrique fournit des informations de référence importantes sur les fichiers de spécification de génération (buildspec). Un buildspec est un ensemble de commandes de construction et de paramètres associés, au format YAML, qui est CodeBuild utilisé pour exécuter un build. Vous pouvez inclure une spécification de construction dans le code source ou vous pouvez définir une spécification de construction lorsque vous créez un projet de construction. Pour plus d'informations sur le fonctionnement des spécifications de génération, consultez [Fonctionnement de CodeBuild](#).

Rubriques

- [Nom de fichier buildspec et emplacement de stockage](#)
- [Syntaxe d'un fichier buildspec](#)
- [Exemple de fichier buildspec](#)
- [Versions de fichier buildspec](#)
- [Référence de la spécification de construction par Batch](#)
- [GitHub Référence de spécification de construction d'Action Runner](#)

Nom de fichier buildspec et emplacement de stockage

Si vous incluez une spécification de build dans le cadre du code source, par défaut, le fichier de spécification de build doit être nommé `buildspec.yml` et être placé à la racine de votre répertoire source.

Vous pouvez remplacer le nom et l'emplacement par défaut du fichier de spécification de build. Par exemple, vous pouvez :

- Utiliser un fichier de spécification de build distinct pour différentes builds dans le même référentiel, comme par exemple `buildspec_debug.yml` et `buildspec_release.yml`.
- Stocker un fichier de spécification de build ailleurs qu'à la racine de votre répertoire source, par exemple `config/buildspec.yml` ou dans un compartiment S3. Le compartiment S3 doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

Vous pouvez spécifier une seule spécification de build pour un projet de build, quel que soit le nom du fichier de spécification de build.

Pour remplacer le nom, l'emplacement, ou les deux, du fichier de spécification de build par défaut, effectuez l'une des actions suivantes :

- Exécutez la `update-project` commande AWS CLI `create-project` or, en définissant la `buildspec` valeur sur le chemin du fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR` Vous pouvez également faire l'équivalent avec l'`create project` opération dans les AWS SDK. Pour plus d'informations, consultez [Création d'un projet de génération](#) ou [Modification des paramètres d'un projet de génération](#).
- Exécutez la AWS CLI `start-build` commande en définissant la `buildspecOverride` valeur sur le chemin du fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR` Vous pouvez également faire l'équivalent avec l'`start build` opération dans les AWS SDK. Pour de plus amples informations, veuillez consulter [Exécution d'une génération](#).
- Dans un AWS CloudFormation modèle, définissez la `BuildSpec` propriété de `Source` in a resource of type `AWS::CodeBuild::Project` sur le chemin d'accès au fichier buildspec alternatif par rapport à la valeur de la variable d'environnement intégrée. `CODEBUILD_SRC_DIR`

Pour plus d'informations, consultez la BuildSpec propriété dans la [source AWS CodeBuild du projet](#) dans le guide de AWS CloudFormation l'utilisateur.

Syntaxe d'un fichier buildspec

Les fichiers de spécification de build doivent être créés au format [YAML](#).

Si une commande contient un caractère ou une chaîne de caractères qui ne sont pas pris en charge par YAML, vous devez placer la commande entre guillemets (""). La commande suivante est placée entre guillemets car un deux-points (:) suivi d'un espace n'est pas autorisé dans YAML. Dans la commande, le guillemet est échappé (\").

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\\\",]//g')"
```

La spécification de build a la syntaxe suivante :

```
version: 0.2

run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
  parameter-store:
    key: "value"
    key: "value"
  exported-variables:
    - variable
    - variable
  secrets-manager:
    key: secret-id:json-key:version-stage:version-id
  git-credential-helper: no | yes

proxy:
  upload-artifacts: no | yes
  logs: no | yes

batch:
```



```
fast-fail: false | true
# build-list:
# build-matrix:
# build-graph:

phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    runtime-versions:
      runtime: version
      runtime: version
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  pre\_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  post\_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
```

```

    - command
  finally:
    - command
    - command
  # steps:
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:
  paths:
    - path
    - path

```

La spécification de build contient les éléments suivants :

version

Mappage obligatoire. Représente la version de la spécification de build. Nous vous recommandons d'utiliser `0.2`.

Note

Bien que la version 0.1 soit toujours prise en charge, nous vous recommandons d'utiliser la version 0.2 dans la mesure du possible. Pour de plus amples informations, veuillez consulter [Versions de fichier buildspec](#).

run-as

Séquence facultative. Disponible uniquement pour les utilisateurs Linux. Spécifie un utilisateur Linux qui exécute des commandes dans ce fichier buildspec. `run-as` accorde à l'utilisateur spécifié des autorisations de lecture et d'exécution. Lorsque vous spécifiez `run-as` en haut du fichier buildspec, il s'applique globalement à toutes les commandes. Si vous ne voulez pas spécifier un utilisateur pour toutes les commandes de fichier buildspec, vous pouvez en spécifier un pour les commandes dans une phase à l'aide de `run-as` dans l'un des blocs phases. Si `run-as` n'est pas spécifié, toutes les commandes sont exécutées en tant qu'utilisateur racine.

env

Séquence facultative. Représente les informations pour une ou plusieurs variables d'environnement personnalisées.

Note

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

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

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

env/ coque

Séquence facultative. Spécifie le shell pris en charge pour les systèmes d'exploitation Linux ou Windows.

Pour les systèmes d'exploitation Linux, les balises shell prises en charge sont les suivantes :

- `bash`
- `/bin/sh`

Pour les systèmes d'exploitation Windows, les balises shell prises en charge sont les suivantes :

- `powershell.exe`
- `cmd.exe`

env/variables

Obligatoire si `env` est spécifié, et que vous voulez définir des variables d'environnement personnalisées en texte brut. Contient un mappage de scalaires *clé / valeur*, où chaque mappage représente une variable d'environnement personnalisée unique en texte brut. *clé* est le nom de la variable d'environnement personnalisée, et *valeur* est la valeur de cette variable.

Important

Nous déconseillons vivement le stockage de valeurs sensibles dans les variables d'environnement. Les variables d'environnement peuvent être affichées en texte brut à l'aide d'outils tels que la CodeBuild console et le AWS CLI. Pour les valeurs sensibles, nous vous recommandons d'utiliser à la place le mappage `parameter-store` ou `secrets-manager`, comme décrit plus loin dans cette section.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la

valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée. Vous pouvez ajouter ou remplacer des variables d'environnement lorsque vous créez une génération. Pour de plus amples informations, veuillez consulter [Exécution d'une génération dans AWS CodeBuild](#).
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités. Vous pouvez ajouter des variables d'environnement au niveau d'un projet lorsque vous créez ou modifiez celui-ci. Pour plus d'informations, consultez [Création d'un projet de génération dans AWS CodeBuild](#) et [Modification des paramètres d'un projet de génération dans AWS CodeBuild](#).
- La valeur figurant dans la déclaration `buildspec` a la priorité la plus faible.

env/magasin de paramètres

Obligatoire si cela `env` est spécifié et si vous souhaitez récupérer des variables d'environnement personnalisées stockées dans Amazon EC2 Systems Manager Parameter Store. Contient un mappage de scalaires *clé/valeur*, où chaque mappage représente une variable d'environnement personnalisée unique stockée dans le magasin de paramètres Amazon EC2 Systems Manager. *key* est le nom que vous utiliserez ultérieurement dans vos commandes de compilation pour faire référence à cette variable d'environnement personnalisée, et *value* est le nom de la variable d'environnement personnalisée stockée dans Amazon EC2 Systems Manager Parameter Store. Pour stocker des valeurs sensibles, consultez la section [Stockage des paramètres de Systems Manager](#) et [procédure pas à pas : création et test d'un paramètre de chaîne \(console\)](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Important

CodeBuild Pour permettre de récupérer des variables d'environnement personnalisées stockées dans Amazon EC2 Systems Manager Parameter Store, vous devez ajouter `ssm:GetParameters` l'action à CodeBuild votre rôle de service. Pour de plus amples informations, veuillez consulter [Création d'un rôle CodeBuild de service](#).

Toutes les variables d'environnement que vous récupérez depuis Amazon EC2 Systems Manager Parameter Store remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée MY_VAR avec une valeur my_value, et que vous récupérez une variable d'environnement nommée MY_VAR avec une valeur other_value, la valeur my_value est alors remplacée par other_value. De même, si l'image Docker contient déjà une variable d'environnement nommée PATH avec une valeur /usr/local/sbin:/usr/local/bin, et que vous récupérez une variable d'environnement nommée PATH avec une valeur \$PATH:/usr/share/ant/bin, la valeur /usr/local/sbin:/usr/local/bin est alors remplacée par la valeur littérale \$PATH:/usr/share/ant/bin.

Ne stockez pas de variables d'environnement avec un nom commençant par CODEBUILD_. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée. Vous pouvez ajouter ou remplacer des variables d'environnement lorsque vous créez une génération. Pour de plus amples informations, veuillez consulter [Exécution d'une génération dans AWS CodeBuild](#).
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités. Vous pouvez ajouter des variables d'environnement au niveau d'un projet lorsque vous créez ou modifiez celui-ci. Pour plus d'informations, consultez [Création d'un projet de génération dans AWS CodeBuild](#) et [Modification des paramètres d'un projet de génération dans AWS CodeBuild](#).
- La valeur figurant dans la déclaration buildspec a la priorité la plus faible.

env/Secrets Manager

Obligatoire si vous souhaitez récupérer des variables d'environnement personnalisées stockées dans AWS Secrets Manager. Spécifiez un Gestionnaire de Secrets `reference-key` en utilisant le modèle suivant :

```
<key>: <secret-id>:<json-key>:<version-stage>:<version-id>
```

```
<key>
```

(Obligatoire) Le nom de la variable d'environnement locale. Utilisez ce nom pour accéder à la variable lors de la génération.

<secret-id>

(Obligatoire) Le nom ou Amazon Resource Name (ARN) qui sert d'identifiant unique pour le secret. Pour accéder à un secret dans votre compte AWS, spécifiez simplement le nom secret. Pour accéder à un secret dans un autre AWS compte, spécifiez l'ARN du secret.

<json-key>

(Facultatif) Spécifie le nom de clé de la paire clé-valeur de Secrets Manager dont vous souhaitez récupérer la valeur. Si vous ne spécifiez pas de `json-key`, CodeBuild récupère l'intégralité du texte secret.

<version-stage>

(Facultatif) Spécifie la version secrète que vous souhaitez récupérer à l'aide de l'étiquette intermédiaire attachée à la version. Les étiquettes intermédiaires sont utilisées pour assurer le suivi des différentes versions au cours du processus de rotation. Si vous utilisez la `version-stage`, ne spécifiez pas `version-id`. Si vous ne spécifiez pas d'étape de version ni d'ID de version, par défaut, vous devez récupérer la version avec la valeur d'étape de version de `AWSCURRENT`.

<version-id>

(Facultatif) Spécifie l'identifiant unique de la version du secret que vous souhaitez utiliser. Si vous spécifiez `version-id`, ne spécifiez pas `version-stage`. Si vous ne spécifiez pas d'étape de version ni d'ID de version, par défaut, vous devez récupérer la version avec la valeur d'étape de version de `AWSCURRENT`.

Dans l'exemple suivant, `TestSecret` c'est le nom de la paire clé-valeur stockée dans Secrets Manager. La clé pour `TestSecret` nous `MY_SECRET_VAR`. Vous pouvez accéder à la variable pendant la construction en utilisant le `LOCAL_SECRET_VAR` nom.

```
env:  
  secrets-manager:  
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

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

env/variables exportées

Mappage facultatif. Permet de répertorier les variables d'environnement que vous souhaitez exporter. Spécifiez le nom de chaque variable que vous souhaitez exporter sur une ligne distincte

sous `exported-variables`. La variable que vous souhaitez exporter doit être disponible dans votre conteneur pendant la génération. La variable que vous exportez peut être une variable d'environnement.

Les variables d'environnement exportées sont utilisées conjointement AWS CodePipeline pour exporter les variables d'environnement de la phase de construction actuelle vers les étapes suivantes du pipeline. Pour plus d'informations, consultez la section [Utilisation des variables](#) dans le Guide de AWS CodePipeline l'utilisateur.

Lors d'une génération, la valeur d'une variable est disponible dès la phase `install`. Elle peut être mise à jour entre le début de la phase `install` et la fin de la phase `post_build`. Lorsque la phase `post_build` est terminée, la valeur des variables exportées ne peut pas changer.

Note

Les éléments suivants ne peuvent pas être exportés :

- Les secrets du magasin de paramètres Amazon EC2 Systems Manager sont spécifiés dans le projet de construction.
- Secrets Manager : secrets spécifiés dans le projet de construction
- Les variables d'environnement qui commencent par `AWS_`.

`env/ git-credential-helper`

Mappage facultatif. Utilisé pour indiquer s'il CodeBuild utilise son assistant d'identification Git pour fournir des informations d'identification Git. `yess` il est utilisé. Sinon, `no` ou non spécifié. Pour plus d'informations, consultez [gitcredentials](#) sur le site web de Git.

Note

`git-credential-helper` n'est pas pris en charge pour les générations qui sont déclenchées par un webhook pour un référentiel Git public.

`proxy`

Séquence facultative. Utilisé pour représenter les paramètres si vous exécutez votre génération dans un serveur proxy explicite. Pour de plus amples informations, veuillez consulter [Exécution de CodeBuild sur un serveur proxy explicite](#).

proxy/charger les artefacts

Mappage facultatif. Définissez ce paramètre sur `yes` si vous souhaitez que votre build dans un serveur proxy explicite charge des artefacts. L'argument par défaut est `no`.

proxy/journaux

Mappage facultatif. Définissez ce `yes` paramètre pour intégrer un serveur proxy explicite afin de créer des CloudWatch journaux. L'argument par défaut est `no`.

phases

Séquence obligatoire. Représente les CodeBuild commandes exécutées pendant chaque phase de la génération.

Note

Dans la version 0.1 de `buildspec`, CodeBuild exécute chaque commande dans une instance distincte du shell par défaut dans l'environnement de construction. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par conséquent, par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes (par exemple, pour le changement de répertoire ou la définition de variables d'environnement). Pour contourner ce problème, nous vous recommandons d'utiliser la version 0.2, qui permet de résoudre ce problème. Si vous devez utiliser la spécification de build de version 0.1, nous vous recommandons les approches figurant dans [Shells et commandes dans les environnements de génération](#).

phases/*/run-as

Séquence facultative. Utilisez-le dans une phase de build pour spécifier un utilisateur Linux qui exécute ses commandes. Si `run-as` est également spécifié globalement pour toutes les commandes en haut du fichier `buildspec`, alors l'utilisateur au niveau de phase est prioritaire. Par exemple, si `run-as` spécifie globalement l'utilisateur-1, et pour la phase `install` seule une instruction `run-as` spécifie l'utilisateur 2, alors toutes les commandes dans le fichier `buildspec` sont exécutées en tant qu'utilisateur-1, sauf les commandes dans la phase `install`, qui sont exécutées en tant qu'utilisateur-2.

phases/*/ en cas de défaillance

Séquence facultative. Spécifie l'action à entreprendre en cas de défaillance au cours de la phase. Il peut s'agir de l'une des valeurs suivantes :

- ABORT- Annulation de la construction.
- CONTINUE- Passez à la phase suivante.

Si cette propriété n'est pas spécifiée, le processus d'échec suit les phases de transition, comme indiqué dans [Transitions des phases de génération](#).

phases/*/ enfin

Bloc facultatif. Les commandes spécifiées dans un `finally` bloc sont exécutées après les commandes du `commands` bloc. Les commandes d'un `finally` bloc sont exécutées même si une commande du `commands` bloc échoue. Par exemple, si le `commands` bloc contient trois commandes et que la première échoue, CodeBuild ignore les deux commandes restantes et exécute toutes les commandes du `finally` bloc. La phase est réussie lorsque toutes les commandes des blocs `commands` et `finally` s'exécutent avec succès. Si une commande d'une phase échoue, la phase échoue.

Les noms de phase de génération autorisés sont :

phases/installer

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées pendant l'installation. Nous vous recommandons d'utiliser la phase `install` uniquement pour l'installation de packages dans l'environnement de génération. Par exemple, vous pouvez utiliser cette phase pour installer une infrastructure de test de code comme Mocha ou RSpec.

phases/installer/versions d'exécution

Séquence facultative. Une version d'exécution est prise en charge avec l'image standard Ubuntu 5.0 ou ultérieure et l'image standard Amazon Linux 2 4.0 ou version ultérieure. Si cette valeur est spécifiée, au moins une exécution doit être incluse dans cette section. Spécifiez un environnement d'exécution utilisant une version spécifique, une version majeure suivie de `.x` pour spécifier que CodeBuild utilise cette version majeure avec sa dernière version mineure, ou `latest` pour utiliser les versions majeure et mineure les plus récentes (par exemple `ruby: 3.2`, `nodejs: 18.x`, ou `java: latest`). Vous pouvez spécifier l'exécution à l'aide d'un nombre ou d'une variable d'environnement. Par exemple, si vous utilisez l'image standard 4.0

d'Amazon Linux 2, ce qui suit indique que la version 17 de Java, la dernière version mineure de python version 3 et une version contenue dans une variable d'environnement de Ruby sont installées. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

Vous pouvez spécifier un ou plusieurs runtimes dans la section `runtime-versions` de votre fichier `buildspec`. Si votre runtime dépend d'un autre runtime, vous pouvez également spécifier son runtime dépendant dans le fichier `buildspec`. Si vous ne spécifiez aucun environnement d'exécution dans le fichier `buildspec`, CodeBuild choisit les environnements d'exécution par défaut disponibles dans l'image que vous utilisez. Si vous spécifiez un ou plusieurs environnements d'exécution, utilisez uniquement CodeBuild ces environnements d'exécution. Si aucun environnement d'exécution dépendant n'est spécifié, CodeBuild tente de le choisir pour vous.

Si deux exécutions spécifiées entrent en conflit, la génération échoue. Par exemple, `android: 29` et `java: openjdk11` sont en conflit, par conséquent si les deux sont spécifiés, la génération échoue.

Pour plus d'informations sur les environnements d'exécution disponibles, consultez [Runtimes disponibles](#).

Note

Si vous spécifiez une `runtime-versions` section et utilisez une image autre qu'Ubuntu Standard Image 2.0 ou version ultérieure, ou l'image standard Amazon Linux 2 (AL2) 1.0 ou version ultérieure, le build émet l'avertissement « `Skipping install of runtimes. Runtime version selection is not supported by this build image.` »

phases/installer/commandes

Séquence facultative. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée pendant l'installation. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/prégénération

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées avant la génération. Par exemple, vous pouvez utiliser cette phase pour vous connecter à Amazon ECR, ou vous pouvez installer des dépendances npm.

phases/prégénération/commandes

Séquence obligatoire si `pre_build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée avant la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/génération

Séquence facultative. Représente les commandes, le cas échéant, CodeBuild exécutées pendant la génération. Par exemple, vous pouvez utiliser cette phase pour exécuter Mocha, RSpec ou sbt.

phases/génération/commandes

Obligatoire si `build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée pendant la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

phases/post-génération

Séquence facultative. Représente les commandes, le cas échéant, qui CodeBuild s'exécutent après la génération. Par exemple, vous pouvez utiliser Maven pour empaqueter les artefacts de construction dans un fichier JAR ou WAR, ou vous pouvez transférer une image Docker dans Amazon ECR. Vous pouvez ensuite envoyer une notification de build via Amazon SNS.

phases/post-génération/commandes

Obligatoire si `post_build` est spécifié. Contient une séquence de scalaires, où chaque scalaire représente une commande unique CodeBuild exécutée après la génération. CodeBuild exécute chaque commande, une par une, dans l'ordre indiqué, du début à la fin.

rapports

report-group-name-or-fil

Séquence facultative. Spécifie le groupe de rapports auquel les rapports sont envoyés. Un projet peut comporter un maximum de cinq groupes de rapports. Spécifiez l'ARN d'un groupe de rapports existant ou le nom d'un nouveau groupe de rapports. Si vous spécifiez un nom, CodeBuild crée un groupe de rapports en utilisant le nom de votre projet et le nom que vous spécifiez dans le format `<project-name>-<report-group-name>`. Le nom du groupe de rapports peut également être défini à l'aide d'une variable d'environnement dans la spécification de construction, telle que `$REPORT_GROUP_NAME`. Pour de plus amples informations, veuillez consulter [Attribution des noms des groupes de rapports](#).

rapports/<groupe de rapports>/fichiers

Séquence obligatoire. Représente les emplacements qui contiennent les données brutes des résultats de test générés par le rapport. Contient une séquence de scalaires, chaque scalaire représentant un emplacement distinct où se CodeBuild trouvent les fichiers de test, par rapport à l'emplacement de construction d'origine ou, s'il est défini, au `base-directory`. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-test-report-file.json`).
- Un fichier unique dans un sous-répertoire (par exemple, `my-subdirectory/my-test-report-file.json` ou `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/*'` représente tous les fichiers de manière récursive.
- `my-subdirectory/*` représente tous les fichiers dans un sous-répertoire nommé *mon-sous-répertoire*.
- `my-subdirectory/**/*` représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé *mon-sous-répertoire*.

rapports/<groupe de rapports>/format de fichier

Mappage facultatif. Représente le format du fichier de rapport. Si non spécifié, JUNITXML est utilisé. Cette valeur ne distingue pas les majuscules et minuscules. Les valeurs possibles sont :

Rapports d'essais

CUCUMBERJSON

Cucumber JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

NUnit 3 XML

TESTNGXML

TestNG XML

VISUALSTUDIOTRX

Visual Studio TRX

Rapports sur la couverture du code

CLOVERXML

Trèfle XML

COBERTURAXML


Couverture XML

JACOCOXML

JaCoCo XML

SIMPLECOV

SimpleCov JSON

 Note

CodeBuild [accepte les rapports de couverture du code JSON générés par simplecov, et non par simplecov-json.](#)

rapports/<groupe de rapports>/répertoire de base

Mappage facultatif. Représente un ou plusieurs répertoires de niveau supérieur, relatifs à l'emplacement de construction d'origine, qui CodeBuild permettent de déterminer où trouver les fichiers de test bruts.

rapports/<groupe de rapports>/annuler les chemins

Facultatif. Spécifie si les répertoires de fichiers de rapport sont aplatis dans la sortie. Si cela n'est pas spécifié ou si `no` est défini, les fichiers de rapport sont générés avec leur structure de répertoire intacte. Si `yes` est défini, tous les fichiers de test sont placés dans le même répertoire de sortie. Par exemple, si un chemin d'accès à un résultat de test est `com/myapp/mytests/TestResult.xml`, la définition de `yes` place ce fichier dans `/TestResult.xml`.

artefacts

Séquence facultative. Représente des informations indiquant où se CodeBuild trouve la sortie de génération et comment CodeBuild la préparer pour le téléchargement vers le compartiment de sortie S3. Cette séquence n'est pas obligatoire si, par exemple, vous créez et transférez une image Docker vers Amazon ECR, ou si vous exécutez des tests unitaires sur votre code source, mais que vous ne le créez pas.

Note

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

artefacts/fichiers

Séquence obligatoire. Représente les emplacements qui contiennent les artefacts de sortie de génération dans l'environnement de génération. Contient une séquence de scalaires, où chaque scalaire représente un emplacement distinct où CodeBuild peut trouver des artefacts de sortie de génération, par rapport à l'emplacement de génération d'origine ou, si défini, au répertoire de base. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-file.jar`).

- Un fichier unique dans un sous-répertoire (par exemple, *my-subdirectory*/my-file.jar ou *my-parent-subdirectory*/*my-subdirectory*/my-file.jar).
- `'**/*'` représente tous les fichiers de manière récursive.
- *my-subdirectory*/* représente tous les fichiers dans un sous-répertoire nommé *mon-sous-répertoire*.
- *my-subdirectory***/* représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé *mon-sous-répertoire*.

Lorsque vous spécifiez les emplacements des artefacts de sortie de construction, vous CodeBuild pouvez localiser l'emplacement de construction d'origine dans l'environnement de construction. Vous n'avez pas besoin de préfixer les emplacements d'artefact de sortie de génération avec le chemin de l'emplacement de génération d'origine, ni de spécifier `./` ou un élément similaire. Si vous souhaitez connaître le chemin d'accès à cet emplacement, vous pouvez exécuter une commande comme `echo $CODEBUILD_SRC_DIR` pendant une génération. L'emplacement de chaque environnement de génération peut être légèrement différent.

artefacts/nom

Nom facultatif. Spécifie un nom pour votre artefact de génération. Ce nom est utilisé lorsque l'une des conditions suivantes est vraie.

- Vous utilisez l' API CodeBuild pour créer vos versions et `overrideArtifactName` est défini sur l'`ProjectArtifacts` objet lorsqu'un projet est mis à jour, qu'un projet est créé ou qu'un build est lancé.
- Vous utilisez la CodeBuild console pour créer vos versions, un nom est spécifié dans le fichier `buildspec` et vous sélectionnez `Activer le versionnement sémantique` lorsque vous créez ou mettez à jour un projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(console\)](#).

Vous pouvez spécifier un nom dans le fichier de spécification de build qui est calculé au moment de la génération. Le nom spécifié dans un fichier de spécification de build utilise le langage de commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour de plus amples informations, veuillez consulter [Langage de commandes Shell](#).

- Ceci est un exemple de nom d'artefact suivi de la date à laquelle l'artefact a été créé.

```
version: 0.2
phases:
  build:
```



```

  commands:
    - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)

```

- Il s'agit d'un exemple de nom d'artefact qui utilise une variable d' CodeBuild environnement. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION

```

- Il s'agit d'un exemple de nom d'artefact qui utilise une variable d' CodeBuild environnement à laquelle est ajoutée la date de création de l'artefact.

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: $AWS_REGION-$(date +%Y-%m-%d)

```

Vous pouvez ajouter des informations de chemin au nom afin que les artefacts nommés soient placés dans des répertoires en fonction du chemin indiqué dans le nom. Dans cet exemple, les artefacts de construction sont placés dans la sortie sousbuilds/<build number>/my-artifacts.

```

version: 0.2
phases:
  build:

```

```
  commands:
    - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: builds/$CODEBUILD_BUILD_NUMBER/my-artifacts
```

artefacts/ignorer les chemins

Facultatif. Spécifie si les répertoires d'artefacts de génération sont aplatis dans la sortie. Si cela n'est pas spécifié, si `no` est défini, les artefacts de génération sont générés avec leur structure de répertoire intacte. Si `yes` est défini, tous les artefacts de génération sont placés dans le même répertoire de sortie. Par exemple, si un chemin d'accès à un fichier dans l'artefact de sortie de génération est `com/mycompany/app/HelloWorld.java`, la définition de `yes` permet de placer ce fichier dans `/HelloWorld.java`.

artefacts/répertoire de base

Mappage facultatif. Représente un ou plusieurs répertoires de niveau supérieur, relatifs à l'emplacement de génération d'origine, qui permettent CodeBuild de déterminer les fichiers et sous-répertoires à inclure dans l'artefact de sortie de génération. Les valeurs valides sont les suivantes :

- Un répertoire unique de niveau supérieur (par exemple, `my-directory`).
- `'my-directory*'` représente tous les répertoires de niveau supérieur avec les noms commençant par `my-directory`.

Les répertoires de niveau supérieur correspondants ne sont pas inclus dans l'artefact de sortie de génération, seulement leurs fichiers et sous-répertoires.

Vous pouvez utiliser `files` et `discard-paths` pour limiter encore plus les fichiers et sous-répertoires inclus. Par exemple, pour la structure de répertoire suivante :

```
.
### my-build-1
#   ### my-file-1.txt
### my-build-2
    ### my-file-2.txt
    ### my-subdirectory
        ### my-file-3.txt
```

Et pour la séquence `artifacts` suivante :

```
artifacts:
  files:
    - '*my-file-3.txt'
  base-directory: my-build-2
```

Le sous-répertoire et le fichier suivants seront inclus dans l'artefact de sortie de génération :

```
.
### my-subdirectory
### my-file-3.txt
```

Alors que pour la séquence `artifacts` suivante :

```
artifacts:
  files:
    - '**/*'
  base-directory: 'my-build*'
  discard-paths: yes
```

Les fichiers suivants seront inclus dans l'artefact de sortie de génération :

```
.
### my-file-1.txt
### my-file-2.txt
### my-file-3.txt
```

artéfacts/ chemins d'exclusion

Mappage facultatif. Représente un ou plusieurs chemins, relatifs à `base-directory`, qui CodeBuild seront exclus de la construction des artefacts. L'astérisque (*) correspond à zéro ou plusieurs caractères d'un composant de nom sans dépasser les limites d'un dossier. Un double astérisque (**) correspond à zéro ou plusieurs caractères d'un composant de nom dans tous les répertoires.

Les exemples de chemins d'exclusion sont les suivants :

- Pour exclure un fichier de tous les répertoires : `**/file-name/**/*`
- Pour exclure tous les dossiers à points, procédez comme suit : `**/*.*/**/*`
- Pour exclure tous les fichiers à points : `**/*.*`

artifacts/ activer les liens symboliques

Facultatif. Si le type de sortie est ZIP, indique si les liens symboliques internes sont conservés dans le fichier ZIP. Si tel est le cas, tous les liens symboliques internes de la source seront conservés dans le fichier ZIP des artefacts.

artifacts/ préfixe s3

Facultatif. Spécifie un préfixe utilisé lorsque les artefacts sont envoyés vers un compartiment Amazon S3 et que le type d'espace de noms est. BUILD_ID Lorsqu'il est utilisé, le chemin de sortie dans le compartiment est `<s3-prefix>/<build-id>/<name>.zip`.

artifacts/artefacts secondaires

Séquence facultative. Représente une ou plusieurs définitions d'artefacts comme correspondance entre l'identificateur d'un artefact et la définition d'un artefact. Chaque identifiant d'artefact de ce bloc doit correspondre à un artefact défini dans l'attribut `secondaryArtifacts` de votre projet. Chaque définition distincte a la même syntaxe que le bloc `artifacts` ci-dessus.

Note

La [artifacts/files](#) séquence est toujours requise, même lorsque seuls des artefacts secondaires sont définis.

Par exemple, si la structure de votre projet est la suivante :

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

```
}
```

Votre fichier buildspec ressemble alors à ceci :

```
version: 0.2

phases:
  build:
    commands:
      - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      files:
        - directory/file1
      name: secondary-artifact-name-1
    artifact2:
      files:
        - directory/file2
      name: secondary-artifact-name-2
```

cache

Séquence facultative. Représente des informations sur l'endroit où CodeBuild préparer les fichiers pour le téléchargement du cache dans un compartiment de cache S3. Cette séquence n'est pas requise si le type de cache du projet est No Cache.

cache/chemins

Séquence obligatoire. Représente les emplacements du cache. Contient une séquence de scalaires, chaque scalaire représentant un emplacement distinct où CodeBuild peuvent trouver les artefacts de sortie de construction, par rapport à l'emplacement de construction d'origine ou, s'il est défini, au répertoire de base. Il peut s'agir notamment des emplacements suivants :

- Un fichier unique (par exemple, `my-file.jar`).
- Un fichier unique dans un sous-répertoire (par exemple, `my-subdirectory/my-file.jar` ou `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` représente tous les fichiers de manière récursive.

- `my-subdirectory/*` représente tous les fichiers dans un sous-répertoire nommé *mon-sous-répertoire*.
- `my-subdirectory/**/*` représente tous les fichiers de manière récursive à partir d'un sous-répertoire nommé *mon-sous-répertoire*.

Important

Comme une déclaration de spécification de build doit être dans un format YAML valide, les espaces sont importants dans une déclaration de spécification de build. Si le nombre d'espaces dans votre déclaration de spécification de build n'est pas valide, les builds peuvent échouer immédiatement. Vous pouvez utiliser un validateur YAML pour tester si vos déclarations de spécification de build sont dans un format YAML valide.

Si vous utilisez le AWS CLI ou les AWS SDK pour déclarer une spécification de construction lorsque vous créez ou mettez à jour un projet de construction, la spécification de construction doit être une chaîne unique exprimée au format YAML, avec les espaces blancs et les caractères d'échappement de nouvelle ligne requis. Vous trouverez un exemple dans la section suivante.

Si vous utilisez les AWS CodePipeline consoles CodeBuild ou au lieu d'un fichier `buildspec.yml`, vous pouvez insérer des commandes pour la phase uniquement. `build` Au lieu d'utiliser la syntaxe précédente, vous répertoriez, sur une seule ligne, toutes les commandes que vous souhaitez exécuter lors de la phase de génération (build). Pour plusieurs commandes, séparez celles-ci avec `&&` (par exemple, `mvn test && mvn package`).

Vous pouvez utiliser les CodePipeline consoles CodeBuild ou au lieu d'un fichier `buildspec.yml` pour spécifier les emplacements des artefacts de sortie de génération dans l'environnement de génération. Au lieu d'utiliser la syntaxe précédente, vous répertoriez, sur une seule ligne, tous les emplacements. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, `buildspec.yml, target/my-app.jar`).

Exemple de fichier buildspec

Voici un exemple de fichier `buildspec.yml`.

```
version: 0.2

env:
```

```
variables:
  JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
parameter-store:
  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`

reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  files:
    - "**/*"
  base-directory: 'target/tests/reports'
  discard-paths: no
  reportGroupCucumberJson:
  files:
    - 'cucumber/target/cucumber-tests.xml'
  discard-paths: yes
  file-format: CUCUMBERJSON # default is JUNITXML

artifacts:
  files:
```

```

- target/messageUtil-1.0.jar
discard-paths: yes
secondary-artifacts:
  artifact1:
    files:
      - target/artifact-1.0.jar
    discard-paths: yes
  artifact2:
    files:
      - target/artifact-2.0.jar
    discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'

```

Voici un exemple de la spécification de construction précédente, exprimée sous forme de chaîne unique, à utiliser avec le ou AWS CLI les SDK. AWS

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-amd64\\\"
parameter-store:\n  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n
phases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n
      - apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This always runs even if the update or install command fails \n
  pre_build:\n    commands:\n      - echo Entered the pre_build phase...\n      - docker login -u User -p $LOGIN_PASSWORD\n
    finally:\n      - echo This always runs even if the login command fails \n
  build:\n    commands:\n      - echo Entered the build phase...\n      - echo Build started on `date`\n      - mvn install\n
    finally:\n      - echo This always runs even if the install command fails\n
  post_build:\n    commands:\n      - echo Entered the post_build phase...\n      - echo Build completed on `date`\n\n  reports:\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\n  artifacts:\n
    artifact1:\n      files:\n        - target/messageUtil-1.0.jar\n
      discard-paths: yes\n
    artifact2:\n      files:\n        - target/messageUtil-1.0.jar\n
      discard-paths: yes\n
  cache:\n    paths:\n      - '/root/.m2/**/*'

```

Voici un exemple des commandes de la build phase, à utiliser avec les CodePipeline consoles CodeBuild or.

```
echo Build started on `date` && mvn install
```


Dans les exemples suivants :

- Une variable d'environnement personnalisée, en texte brut, avec la clé `JAVA_HOME` et la valeur `/usr/lib/jvm/java-8-openjdk-amd64` est définie.
- Une variable d'environnement personnalisée nommée `dockerLoginPassword` stockée dans Amazon EC2 Systems Manager Parameter Store est référencée ultérieurement dans les commandes de construction à l'aide de la `LOGIN_PASSWORD` clé.
- Vous ne pouvez pas modifier ces noms de phase de génération. Les commandes exécutées dans cet exemple sont `apt-get update -y` et `apt-get install -y maven` (pour installer Apache Maven), `mvn install` (pour compiler, tester et empaqueter le code source dans un artefact de sortie de build et pour installer l'artefact de sortie de build dans son référentiel interne), `docker login` (pour se connecter à Docker avec le mot de passe correspondant à la valeur de la variable d'environnement personnalisée que `dockerLoginPassword` vous avez définie dans Amazon EC2 Systems Manager Parameter Store) et plusieurs commandes. Les commandes `echo` sont incluses ici pour montrer comment les CodeBuild commandes sont exécutées et l'ordre dans lequel elles sont exécutées.
- `files` représente les fichiers à charger dans l'emplacement de sortie de génération. Dans cet exemple, CodeBuild télécharge le fichier `messageUtil-1.0.jar` unique. Le fichier `messageUtil-1.0.jar` peut être trouvé dans le répertoire relatif nommé `target` dans l'environnement de génération. Comme `discard-paths: yes` est spécifié, `messageUtil-1.0.jar` est chargé directement (et non dans un répertoire `target` intermédiaire). Le nom de fichier `messageUtil-1.0.jar` et le nom de répertoire relatif de `target` sont basés sur la façon dont Apache crée et stocke les artefacts de sortie de génération pour cet exemple uniquement. Dans vos propres scénarios, ces noms de fichier et ces répertoires seront différents.
- `reports` représente deux groupes de rapports qui génèrent des rapports pendant la génération :
 - `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` spécifie l'ARN d'un groupe de rapports. Les résultats des tests générés par le framework de test se trouvent dans le répertoire `target/tests/reports`. Le format de fichier est `JUnitXml` et le chemin d'accès n'est pas supprimé des fichiers contenant les résultats du test.
 - `reportGroupCucumberJson` spécifie un nouveau groupe de rapports. Si le nom du projet est `my-project`, un groupe de rapports portant le nom `my-project-reportGroupCucumberJson` est créé lors de l'exécution d'une génération. Les résultats des tests générés par le framework de test sont dans `cucumber/target/cucumber-tests.xml`.

Le format de fichier test est `CucumberJson` et le chemin d'accès est supprimé des fichiers contenant les résultats du test.

Versions de fichier buildspec

Le tableau suivant répertorie les versions de spécification de build et les modifications entre les versions.

| Version | Modifications |
|---------|---|
| 0.2 | <ul style="list-style-type: none">• <code>environment_variables</code> a été renommée en <code>env</code>.• <code>plaintext</code> a été renommée en <code>variables</code>.• La propriété <code>type</code> pour <code>artifacts</code> est obsolète.• Dans la version 0.1, AWS CodeBuild exécute chaque commande de construction dans une instance distincte du shell par défaut dans l'environnement de construction. Dans la version 0.2, CodeBuild exécute toutes les commandes de construction dans la même instance du shell par défaut dans l'environnement de construction. |
| 0.1 | Il s'agit de la définition initiale du format de spécification de génération. |

Référence de la spécification de construction par Batch

Cette rubrique contient la référence Buildspec pour les propriétés de génération par lots.

lot

Mappage facultatif. Les paramètres de génération par lots pour le projet.

lotfast-fail

Facultatif. Spécifie le comportement de la génération par lots en cas d'échec d'une ou de plusieurs tâches de génération.

`false`

La valeur par défaut. Toutes les versions en cours seront terminées.

`true`

Toutes les versions en cours seront arrêtées lorsque l'une des tâches de génération échoue.

Par défaut, toutes les tâches de génération par lots s'exécutent avec les paramètres de génération tels que `env`, `phases`, spécifié dans le fichier `buildspec`. Vous pouvez remplacer les paramètres de génération par défaut en spécifiant différents paramètres de génération `env` ou un fichier de spécification de construction différent dans le `batch/<batch-type>/buildspec` Paramètre .

Contenu du document `batch` varie en fonction du type de génération par lot qui est spécifié. Les types de génération par lots possibles sont les suivants :

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)

batch/build-graph

Définit une génération d'un graphe. Un graphique de génération définit un ensemble de tâches qui dépendent d'autres tâches du lot. Pour plus d'informations, consultez [Construire un graphe](#).

Cet élément contient un ensemble de tâches de génération. Chaque tâche de génération contient les propriétés suivantes.

identificateur

Obligatoire. Identifiant de la tâche.

Buildspec

Facultatif. Chemin et nom du fichier `buildspec` à utiliser pour cette tâche. Si ce paramètre n'est pas spécifié, le fichier `buildspec` actuel est utilisé.

session de débogage

Facultatif. Une valeur booléenne qui indique si le débogage de session est activé pour cette génération par lot. Pour de plus amples informations sur le débogage de session, veuillez consulter [Afficher une version en cours d'exécution dans le Gestionnaire de session](#).

`false`

Le débogage de session est désactivé.

`true`

Le débogage de session est activé.

dépend

Facultatif. Un ensemble d'identificateurs de tâches dont dépend cette tâche. Cette tâche ne sera pas exécutée tant que ces tâches ne seront pas terminées.

env

Facultatif. L'environnement de génération remplace la tâche. Il peut contenir les propriétés suivantes :

type de calcul

Identificateur du type de calcul à utiliser pour la tâche. Voir `computeType` dans [the section called "Modes et types de calcul de l'environnement de création"](#) pour connaître les valeurs possibles.

image

Identifiant de l'image à utiliser pour la tâche. Voir `Identificateur de l'image` dans [the section called "Images Docker fournies par CodeBuild"](#) pour connaître les valeurs possibles.

mode privilégié

Une valeur booléenne qui indique s'il convient d'exécuter le démon Docker dans un conteneur Docker. Définis sur `true` uniquement si le projet de génération est utilisé pour générer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est `false`.

type

Identificateur du type d'environnement à utiliser pour la tâche. Voir `Type d'environnement` dans [the section called "Modes et types de calcul de l'environnement de création"](#) pour connaître les valeurs possibles.

variables

Les variables d'environnement qui seront présentes dans l'environnement de construction.
Pour plus d'informations, consultez [env/variables](#).

ignore-échec

Facultatif. Valeur booléenne qui indique si un échec de cette tâche de génération peut être ignoré.

`false`

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

`true`

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

L'exemple suivant est celui d'une entrée buildspec du graphe de génération :

```
batch:
  fast-fail: false
  build-graph:
    - identifiant: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifiant: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifiant: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
```

batch/build-list

Définit un liste de construction. Une liste de génération permet de définir un certain nombre de tâches exécutées en parallèle. Pour plus d'informations, consultez [Construire une liste](#).

Cet élément contient un ensemble de tâches de génération. Chaque tâche de génération contient les propriétés suivantes.

identificateur

Obligatoire. Identifiant de la tâche.

Buildspec

Facultatif. Chemin et nom du fichier buildspec à utiliser pour cette tâche. Si ce paramètre n'est pas spécifié, le fichier buildspec actuel est utilisé.

session de débogage

Facultatif. Une valeur booléenne qui indique si le débogage de session est activé pour cette génération par lot. Pour de plus amples informations sur le débogage de session, veuillez consulter [Afficher une version en cours d'exécution dans le Gestionnaire de session](#).

false

Le débogage de session est désactivé.

true

Le débogage de session est activé.

env

Facultatif. L'environnement de génération remplace la tâche. Il peut contenir les propriétés suivantes :

type de calcul

Identificateur du type de calcul à utiliser pour la tâche. Voir `computeType` dans [the section called "Modes et types de calcul de l'environnement de création"](#) pour connaître les valeurs possibles.

image

Identifiant de l'image à utiliser pour la tâche. Voir `Identificateur de l'image` dans [the section called "Images Docker fournies par CodeBuild"](#) pour connaître les valeurs possibles.

mode privilégié

Une valeur booléenne qui indique s'il convient d'exécuter le démon Docker dans un conteneur Docker. Définis sur `true` uniquement si le projet de génération est utilisé pour générer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est `false`.

type

Identificateur du type d'environnement à utiliser pour la tâche. Voir `Type d'environnement` dans [the section called "Modes et types de calcul de l'environnement de création"](#) pour connaître les valeurs possibles.

variables

Les variables d'environnement qui seront présentes dans l'environnement de construction. Pour plus d'informations, consultez [env/variables](#).

ignore-échec

Facultatif. Valeur booléenne qui indique si un échec de cette tâche de génération peut être ignoré.

false

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

true

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

L'exemple suivant est celui d'une entrée `buildspec` de liste de génération :

```
batch:
  fast-fail: false
  build-list:
    - identifiant: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifiant: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
```

```
ignore-failure: true
```

batch/build-matrix

Définit une matrice de génération. Une matrice de génération définit les tâches avec différentes configurations exécutées en parallèle. CodeBuild crée une version distincte pour chaque combinaison de configuration possible. Pour plus d'informations, consultez [Construire une matrice](#).

statique

Les propriétés statiques s'appliquent à toutes les tâches de génération.

ignore-échec

Facultatif. Valeur booléenne qui indique si un échec de cette tâche de génération peut être ignoré.

false

La valeur par défaut. Si cette tâche de génération échoue, la génération par lots échouera.

true

Si cette tâche de génération échoue, la génération par lots peut toujours réussir.

env

Facultatif. L'environnement de génération remplace toutes les tâches.

mode privilégié

Une valeur booléenne qui indique s'il convient d'exécuter le démon Docker dans un conteneur Docker. Définis sur `true` uniquement si le projet de génération est utilisé pour générer des images Docker. À défaut, une génération qui tente d'interagir avec le démon Docker échoue. Le paramètre par défaut est `false`.

type

Identificateur du type d'environnement à utiliser pour la tâche. Voir [Type d'environnement](#) dans [the section called "Modes et types de calcul de l'environnement de création"](#) pour connaître les valeurs possibles.

dynamique

Les propriétés dynamiques définissent la matrice de génération.

Buildspec

Facultatif. Tableau contenant le chemin d'accès et les noms de fichiers des fichiers Buildspec à utiliser pour ces tâches. Si ce paramètre n'est pas spécifié, le fichier buildspec actuel est utilisé.

env

Facultatif. L'environnement de génération remplace ces tâches.

type de calcul

Tableau contenant les identificateurs des types de calcul à utiliser pour ces tâches.

Voir `computeType` dans [the section called “Modes et types de calcul de l'environnement de création”](#) pour connaître les valeurs possibles.

image

Tableau contenant les identificateurs des images à utiliser pour ces tâches.

Voir `imageIdentifier` de l'image dans [the section called “Images Docker fournies par CodeBuild”](#) pour connaître les valeurs possibles.

variables

Tableau contenant les variables d'environnement qui seront présentes dans les environnements de génération pour ces tâches. Pour plus d'informations, consultez [env/variables](#).

L'exemple suivant est celui d'une entrée buildspec de génération de matrice de génération :

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Pour de plus amples informations, veuillez consulter [Construire une matrice](#).

GitHubRéférence de spécification de construction d'Action Runner

Cette rubrique contient la référence buildspec pour les propriétés GitHub d'Action Runner.

steps

Séquence facultative. Les étapes sont utilisées pour exécuter des commandes et des actions dansCodeBuild. Pour plus d'informations, veuillez consulter [GitHubAction Runner dansAWS CodeBuild](#).

Note

Chaque phase de construction prend en charge une liste de commands ou une liste de steps, mais les deux ne peuvent pas être utilisées dans la même phase.

Chaque étape de génération contient les propriétés suivantes.

id

Facultatif. L'identifiant de l'étape qui peut être utilisé pour référencer l'étape dans d'autres [contextes](#).

if

Facultatif. Une instruction conditionnelle qui peut être utilisée pour empêcher l'exécution d'une étape à moins qu'une condition ne soit remplie. Cette instruction peut utiliser n'importe quel [contexte](#) pris en charge, tel que le référencement de variables environnementales à partir deCodeBuild, ainsi que des [expressions](#).

nom

Facultatif. Le nom de l'étape. Si le nom n'est pas spécifié, le nom sera par défaut le texte spécifié dans la `run` commande.

utilise

L'action qui s'exécute pour l'étape. Certaines actions nécessitent que vous définissiez des entrées à l'aide de `with`. Consultez le fichier README de l'action pour déterminer les entrées requises. Pour plus d'informations, veuillez consulter [LequelGitHubLes actions que je peux utiliser ?](#).

S'il `uses` est spécifié dans votre phase de construction, il ne peut pas être utilisé avec `run`.

 Note

Il est recommandé d'inclure la version de l'action que vous utilisez. Cela peut être fait en spécifiant une balise Git ref, SHA ou Docker. Pour plus d'informations, voir la syntaxe [steps.uses](#).

`courir`

Commande qui exécute des programmes en ligne de commande. Il peut s'agir de commandes d'une seule ligne ou de commandes multilignes. Par défaut, ces commandes s'exécutent à l'aide d'interpréteurs de commandes ne permettant pas de connexion. Pour choisir une coque différente, utilisez `shell`.

S'il `run` est spécifié dans votre phase de construction, il ne peut pas être utilisé avec `uses`.

`coque`

Facultatif. Le shell spécifié pour cette séquence. Pour les paramètres de shell pris en charge, consultez [steps.shell](#). S'il n'est pas spécifié, le shell utilisé est `bash`. Si `bash` n'est pas disponible, `sh` est utilisé.

`avec`

Facultatif. Une carte des paramètres d'entrée définis par l'action. Chaque paramètre est une paire clé/valeur.

`avec .args`

Facultatif. Chaîne qui définit les entrées d'un conteneur Docker.

`avec .entrypoint`

Facultatif. Le point d'entrée Docker spécifié pour le Dockerfile.

`env`

Facultatif. Les variables spécifiées pour les étapes à utiliser dans l'environnement.

`continue-on-error`

Facultatif. Une valeur booléenne qui indique si un échec de cette séquence d'étapes peut être ignoré.

`false`

La valeur par défaut. Si cette séquence d'étapes échoue, la génération échouera.

`true`

Si cette séquence d'étapes échoue, la génération peut toujours réussir.

délai d'expiration (minutes)

Facultatif. Le nombre maximum de minutes pendant lequel l'étape peut s'exécuter avant de se terminer. Par défaut, il n'y a pas de délai d'expiration. Si le délai d'expiration de l'étape dépasse le délai de génération, l'étape s'arrête lorsque le délai de génération est atteint.

Voici un exemple utilisant l'action [Super-Linter](#) GitHub :

```
version: 0.2
phases:
  build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v5
        env:
          VALIDATE_ALL_CODEBASE: true
          USE_FIND_ALGORITHM: true
          FILTER_REGEX_INCLUDE: '/github/workspace/buildspec.yml'
```

Référence sur les environnements de génération pour AWS CodeBuild

Lorsque vous appelez AWS CodeBuild pour exécuter une génération, vous devez fournir des informations sur l'environnement de génération. Un environnement de génération représente une combinaison de système d'exploitation, d'exécution du langage de programmation et d'outils CodeBuild utilisés pour exécuter une génération. Pour plus d'informations sur le fonctionnement d'un environnement de génération, consultez [Fonctionnement de CodeBuild](#).

Un environnement de génération contient une image Docker. Pour plus d'informations, consultez [le glossaire Docker](#) sur le site web Docker Docs.

Lorsque vous fournissez à CodeBuild des informations sur l'environnement de génération, vous spécifiez l'identificateur d'une image Docker dans un type de référentiel pris en charge. Il s'agit

notamment du référentiel d'images CodeBuild Docker, des images accessibles au public dans Docker Hub et des référentiels Amazon Elastic Container Registry (Amazon ECR) auxquels votre AWS compte est autorisé à accéder.

- Nous vous recommandons d'utiliser les images Docker stockées dans le référentiel d'images Docker CodeBuild, car celles-ci sont optimisées pour être utilisées avec le service. Pour plus d'informations, veuillez consulter [Images Docker fournies par CodeBuild](#).
- Pour obtenir l'identificateur d'une image Docker disponible publiquement dans Docker Hub, veuillez consulter [Searching for Repositories](#) sur le site web Docker Docs.
- Pour savoir comment utiliser les images Docker stockées dans les référentiels Amazon ECR de votre AWS compte, consultez [Échantillon Amazon ECR](#).

En plus d'un identificateur d'image Docker, vous spécifiez également un ensemble de ressources informatiques qui sont utilisées par l'environnement de génération. Pour plus d'informations, veuillez consulter [Modes et types de calcul de l'environnement de création](#).

Rubriques

- [Images Docker fournies par CodeBuild](#)
- [Modes et types de calcul de l'environnement de création](#)
- [Shells et commandes dans les environnements de génération](#)
- [Variables d'environnement dans les environnements de génération](#)
- [Tâches en arrière-plan dans les environnements de génération](#)

Images Docker fournies par CodeBuild

Une image prise en charge est la dernière version majeure d'une image disponible dans CodeBuild et est mise à jour avec des mises à jour mineures et des correctifs. CodeBuild optimise la durée de mise en service des versions avec des images prises en charge en les mettant en cache dans l'[Amazon Machine Images \(AMI\) de la machine](#). Si vous souhaitez bénéficier de la mise en cache et minimiser la durée de mise en service de votre build, sélectionnez Toujours utiliser la dernière image pour cette version d'exécution dans la section Version image de la CodeBuild console au lieu d'une version plus détaillée, telle que `aws/codebuild/amazonlinux2-x86_64-standard:4.0-1.0.0`

CodeBuild met fréquemment à jour la liste des images Docker pour ajouter les dernières images et déprécier les anciennes images. Pour obtenir la liste la plus récente, effectuez l'une des actions suivantes :

- Dans la CodeBuild console, dans l'assistant de création de projet ou sur la page Modifier le projet de génération, pour Image d'environnement, choisissez Image gérée. Effectuez vos sélections dans les listes déroulantes Système d'exploitation, Version d'exécution et Version. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).
- Pour l'interface AWS CLI, exécutez la commande `list-curated-environment-images` :

```
aws codebuild list-curated-environment-images
```

- Pour les kits SDK AWS, appelez l'opération `ListCuratedEnvironmentImages` pour votre langage de programmation cible. Pour plus d'informations, consultez le [Référence des kits SDK et des outils AWS](#).

L'image de base de la plateforme Windows Server Core 2019 n'est disponible que dans les régions suivantes :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Oregon)
- Europe (Irlande)

images de calcul EC2

AWS CodeBuild prend en charge les images Docker suivantes qui sont disponibles pour le calcul EC2 dans CodeBuild

| Plateforme | Identificateur de l'image | Définition |
|-------------------|---|----------------------------------|
| Amazon Linux 2 | <code>aws/codebuild/amazonlinux2-x86_64-standard:4.0</code> | al2/standard/4.0 |
| Amazon Linux 2023 | <code>aws/codebuild/amazonlinux2-x86_64-standard:5.0</code> | al2/standard/5.0 |

| Plateforme | Identificateur de l'image | Définition |
|--------------------------|---|--|
| Amazon Linux 2 | aws/codebuild/amazonlinux2-x86_64-standard:corretto8 | al2/standard/corretto8 |
| Amazon Linux 2 | aws/codebuild/amazonlinux2-x86_64-standard:corretto11 | al2/standard/corretto11 |
| Amazon Linux 2 | aws/codebuild/amazonlinux2-aarch64-standard:2.0 | al2/aarch64/standard/2.0 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux2-aarch64-standard:3.0 | al2/aarch64/standard/3.0 |
| Ubuntu 20.04 | aws/codebuild/standard:5.0 | Ubuntu/standard/5.0 |
| Ubuntu 22.04 | aws/codebuild/standard:6.0 | Ubuntu/standard/6.0 |
| Ubuntu 22.04 | aws/codebuild/standard:7.0 | Ubuntu/standard/7.0 |
| Windows Server Core 2019 | aws/codebuild/windows-base:2019-1.0 | N/A |
| Windows Server Core 2019 | aws/codebuild/windows-base:2019-2.0 | N/A |
| Windows Server Core 2019 | aws/codebuild/windows-base:2019-3.0 | N/A |
| Windows Server Core 2022 | aws/codebuild/windows-base:2022-1.0 | N/A |

Images de calcul Lambda

AWS CodeBuild prend en charge les images Docker suivantes qui sont disponibles pour le AWS Lambda calcul. CodeBuild

Architecture d'**aarch64**

| Plateforme | Identificateur de l'image | Définition |
|-------------------|--|--|
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6 | al-lambda/aarch64/dotnet6 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21 | al-lambda/aarch64/go1.21 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11 | al-lambda/aarch64/corretto11 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17 | al-lambda/aarch64/corretto17 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21 | al-lambda/aarch64/corretto21 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18 | al-lambda/aarch64/nodejs18 |

| Plateforme | Identificateur de l'image | Définition |
|-------------------|--|--|
| Amazon Linux 2023 | aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20 | al-lambda/aarch64/nodejs20 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11 | al-lambda/aarch64/python3.11 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12 | al-lambda/aarch64/python3.12 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2 | al-lambda/aarch64/ruby3.2 |

Architecture d'**x86_64**

| Plateforme | Identificateur de l'image | Définition |
|----------------|--|---|
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6 | al-lambda/x86_64/dotnet6 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21 | al-lambda/x86_64/go1.21 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lam | al-lambda/x86_64/corretto11 |

| Plateforme | Identificateur de l'image | Définition |
|-------------------|---|---|
| | bda-standard:corretto11 | |
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17 | al-lambda/x86_64/corretto17 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21 | al-lambda/x86_64/corretto21 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18 | al-lambda/x86_64/nodejs18 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20 | al-lambda/x86_64/nodejs20 |
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11 | al-lambda/x86_64/python3.11 |
| Amazon Linux 2023 | aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12 | al-lambda/x86_64/python3.12 |

| Plateforme | Identificateur de l'image | Définition |
|----------------|--|--|
| Amazon Linux 2 | aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2 | al-lambda/x86_64/ruby3.2 |

Images déconseillées

Une image obsolète est une image qui n'est plus mise en cache ni mise à jour par CodeBuild. Une image obsolète ne reçoit plus de mises à jour de version mineures ni de mises à jour de version de correctif, et comme elles ne sont plus mises à jour, leur utilisation peut ne pas être sécurisée. Si votre CodeBuild projet est configuré pour utiliser une ancienne version d'image, le processus de provisionnement téléchargera cette image Docker et l'utilisera pour créer l'environnement d'exécution conteneurisé, ce qui peut augmenter la durée de provisionnement et la durée globale de construction.

CodeBuild a rendu obsolètes les images Docker suivantes. Vous pouvez toujours utiliser ces images, mais elles ne seront pas mises en cache sur l'hôte de compilation et augmenteront les délais de provisionnement.

| Plateforme | Identificateur de l'image | Définition | Date d'obsolescence |
|----------------|---|--------------------------|---------------------|
| Amazon Linux 2 | aws/codebuild/amazonlinux2-x86_64-standard:3.0 | al2/standard/3.0 | 9 mai 2023 |
| Ubuntu 18.04 | aws/codebuild/standard:4.0 | ubuntu/standard/4.0 | 31 mars 2023 |
| Amazon Linux 2 | aws/codebuild/amazonlinux2-aarch64-standard:1.0 | al2/aarch64/standard/1.0 | 31 mars 2023 |

| Plateforme | Identificateur de l'image | Définition | Date d'obsolescence |
|----------------|--|---------------------|---------------------|
| Ubuntu 18.04 | aws/codebuild/standard:3.0 | ubuntu/standard/3.0 | 30 juin 2022 |
| Amazon Linux 2 | aws/codebuild/amazonlinux2-x86_64-standard:2.0 | al2/standard/2.0 | 30 juin 2022 |

Rubriques

- [Runtimes disponibles](#)
- [Versions d'environnement d'exécution](#)

Runtimes disponibles

Vous pouvez spécifier un ou plusieurs runtimes dans la section `runtime-versions` de votre fichier `buildspec`. Si votre runtime dépend d'un autre runtime, vous pouvez également spécifier son runtime dépendant dans le fichier `buildspec`. Si vous ne spécifiez aucun environnement d'exécution dans le fichier `buildspec`, CodeBuild choisit les environnements d'exécution par défaut disponibles dans l'image que vous utilisez. Si vous spécifiez un ou plusieurs environnements d'exécution, utilisez uniquement CodeBuild ces environnements d'exécution. Si aucun environnement d'exécution dépendant n'est spécifié, CodeBuild tente de le choisir pour vous. Pour plus d'informations, consultez [Specify runtime versions in the buildspec file](#).

Rubriques

- [Temps d'exécution des images Linux](#)
- [Temps d'exécution des images Windows](#)

Temps d'exécution des images Linux

Le tableau suivant contient les environnements d'exécution disponibles et les images Linux standard qui les prennent en charge.

Runtimes des plateformes Ubuntu et Amazon Linux

| Nom de l'exécution | Version | Images |
|--------------------|--|--|
| dotnet | 3.1 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 5.0 | Norme Ubuntu : 5.0 |
| | 6.0 | Amazon Linux 2 x86_64 Norme Lambda : dotnet6 |
| | | Amazon Linux 2 aArch64 Norme Lambda : dotnet6 |
| | | Amazon Linux 2 x86_64 standard : 4.0 |
| 8.0 | Amazon Linux 2023 x86_64 standard : 5.0 | |
| | Amazon Linux 2023 AARch64 standard : 3.0 | |
| | Norme Ubuntu : 6.0 Norme Ubuntu : 7.0 | |
| golang | 1.12 | Amazon Linux 2 AARch64 standard : 2.0 |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| | 1.13 | Amazon Linux 2 AARch64 standard : 2.0 |
| | 1.14 | Amazon Linux 2 AARch64 standard : 2.0 |
| | 1.15 | Norme Ubuntu : 5.0 |
| | 1.16 | Norme Ubuntu : 5.0 |
| | 1,18 | Amazon Linux 2 x86_64 standard : 4.0 Norme Ubuntu : 6.0 |
| | 1,20 | Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 7.0 |
| | 1,21 | Amazon Linux 2 x86_64 norme Lambda : go1.21 Amazon Linux 2 aArch64 Lambda standard : go1.21 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 7.0 |

| Nom de l'exécution | Version | Images |
|--------------------|------------|--|
| java | corretto8 | Amazon Linux 2 x86_64 standard : corretto8 Amazon Linux 2 AArch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | corretto11 | Amazon Linux 2 x86_64 standard : corretto11 Amazon Linux 2 x86_64 Norme Lambda : corretto11 Amazon Linux 2 AArch64 Lambda standard : Corretto11 Amazon Linux 2 AArch64 standard : 2.0 Norme Ubuntu : 5.0 |

| Nom de l'exécution | Version | Images |
|--------------------|------------|--|
| | corretto17 | <p>Amazon Linux 2 x86_64 Norme Lambda : corretto17</p> <p>Amazon Linux 2 aArch64 Lambda standard : Corretto17</p> <p>Amazon Linux 2 x86_64 standard : 4.0</p> <p>Amazon Linux 2023 x86_64 standard : 5.0</p> <p>Amazon Linux 2023 AArch64 standard : 3.0</p> <p>Norme Ubuntu : 6.0</p> <p>Norme Ubuntu : 7.0</p> |
| | corretto21 | <p>Amazon Linux 2 x86_64 Norme Lambda : corretto21</p> <p>Amazon Linux 2 AArch64 Lambda standard : corretto21</p> <p>Amazon Linux 2023 x86_64 standard : 5.0</p> <p>Amazon Linux 2023 AArch64 standard : 3.0</p> <p>Norme Ubuntu : 7.0</p> |
| nodejs | 10 | <p>Amazon Linux 2 AArch64 standard : 2.0</p> |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| | 12 | Amazon Linux 2 AArch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 14 | Norme Ubuntu : 5.0 |
| | 16 | Amazon Linux 2 x86_64 standard : 4.0 Norme Ubuntu : 6.0 |
| | 18 | Amazon Linux 2 x86_64 Norme Lambda : nodejs18 Amazon Linux 2 AArch64, norme Lambda : nodejs18 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0 |
| | 20 | Amazon Linux 2 x86_64 Norme Lambda : nodejs20 Amazon Linux 2 AArch64, norme Lambda : nodejs20 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0 |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| php | 7.3 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 7.4 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 8.0 | Norme Ubuntu : 5.0 |
| | 8.1 | Amazon Linux 2 x86_64 standard : 4.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 6.0 |
| | 8.2 | Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 7.0 |
| | 8.3 | Amazon Linux 2023 AARch64 standard : 3.0 |
| python | 3.7 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| | 3.8 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 3.9 | Amazon Linux 2 x86_64 standard : 4.0 Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 3,10 | Norme Ubuntu : 6.0 |
| | 3,11 | Amazon Linux 2 x86_64 Norme Lambda : python3.11 Amazon Linux 2 AARch64 Lambda standard : python3.11 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 7.0 |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| | 3,12 | Amazon Linux 2 x86_64 Norme Lambda : python3.12 Amazon Linux 2 AARch64 Lambda standard : python3.12 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AARch64 standard : 3.0 Norme Ubuntu : 7.0 |
| ruby | 2.6 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 2.7 | Amazon Linux 2 AARch64 standard : 2.0 Norme Ubuntu : 5.0 |
| | 3.1 | Amazon Linux 2 x86_64 standard : 4.0 Norme Ubuntu : 6.0 |

| Nom de l'exécution | Version | Images |
|--------------------|---------|---|
| | 3.2 | Amazon Linux 2 x86_64 Norme Lambda : ruby3.2 Amazon Linux 2 AArch64 Lambda standard : ruby3.2 Amazon Linux 2023 x86_64 standard : 5.0 Amazon Linux 2023 AArch64 standard : 3.0 Norme Ubuntu : 7.0 |

Temps d'exécution des images Windows

L'image de base de Windows Server Core 2019 contient les environnements d'exécution suivants.

Runtimes de la plateforme Windows

| Nom de l'exécution | Windows Server Core 2019 standard : versions 1.0 | Windows Server Core 2019 standard : versions 2.0 | Windows Server Core 2019 standard : versions 3.0 |
|--------------------|--|--|--|
| dotnet | 3.1 | 3.1 | 6.0 |
| | 5.0 | 6.0 | 7.0 |
| | | 7.0 | 8.0 |
| SDK Dotnet | 3.1 | 3.1 | 8.0 |
| | 5.0 | 6.0 | |
| | | 7.0 | |
| golang | 1.14 | 1,18 | 1,21 |
| gradle | 6.7 | 7.6 | 8,5 |

| Nom de l'exécution | Windows Server Core 2019 standard : versions 1.0 | Windows Server Core 2019 standard : versions 2.0 | Windows Server Core 2019 standard : versions 3.0 |
|--------------------|--|--|--|
| java | Corretto 11 | Corretto 11 Corretto 17 | Corretto 21 |
| maven | 3.6 | 3.8 | 3.9 |
| nodejs | 14,15 | 16,19 | 20,11 |
| php | 7.4 | 8.1 | 8.3 |
| powershell | 7.1 | 7.2 | 7.4 |
| python | 3.8 | 3,10 | 3,12 |
| ruby | 2.7 | 3.1 | 3.3 |

Versions d'environnement d'exécution

Lorsque vous spécifiez un environnement d'exécution dans la section [runtime-versions](#) de votre fichier buildspec, vous pouvez spécifier une version spécifique, une version majeure spécifique et la dernière version mineure, ou la dernière version. Le tableau suivant répertorie les environnements d'exécution disponibles et décrit comment les spécifier. Les versions d'exécution ne sont pas toutes disponibles sur toutes les images. La sélection de la version d'exécution n'est pas non plus prise en charge pour les images personnalisées. Pour plus d'informations, consultez [Runtimes disponibles](#).

Versions d'exécution des plateformes Ubuntu et Amazon Linux 2

| Nom de l'exécution | Version | Version spécifique | Version spécifique majeure et dernière version mineure | Dernière version |
|--------------------|---------|--------------------|--|------------------|
| android | 28 | android: 28 | android: 28.x | android: latest |
| | 29 | android: 29 | android: 29.x | |

| Nom de l'exécution | Version | Version spécifique | Version spécifique majeure et dernière version mineure | Dernière version |
|--------------------|------------|--------------------|--|------------------|
| 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 | | |
| java | corretto8 | java: corretto | java: corretto .x | java: latest |
| | corretto11 | java: corretto 1 | java: corretto 1.x | |
| | corretto17 | java: corretto 7 | java: corretto 7.x | |
| | corretto21 | java: corretto 1 | java: corretto 1.x | |

| Nom de l'exécution | Version | Version spécifique | Version spécifique majeure et dernière version mineure | Dernière version |
|--------------------|---------|--------------------|--|------------------|
| 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 | |
| php | 7.3 | php: 7.3 | php: 7.x | php: latest |
| | 7.4 | php: 7.4 | | |
| | 8.0 | php: 8.0 | php: 8.x | |
| | 8.1 | php: 8.1 | | |
| | 8.2 | php: 8.2 | | |
| | 8.3 | php: 8.3 | | |
| python | 3.7 | python: 3.7 | python: 3.x | python: latest |
| | 3.8 | python: 3.8 | | |
| | 3.9 | python: 3.9 | | |
| | 3,10 | python: 3.10 | | |
| | 3,11 | python: 3.11 | | |
| | 3,12 | python: 3.12 | | |

| Nom de l'exécution | Version | Version spécifique | Version spécifique majeure et dernière version mineure | Dernière version |
|--------------------|---------|--------------------|--|------------------|
| ruby | 2.6 | ruby: 2.6 | ruby: 2.x | ruby: latest |
| | 2.7 | ruby: 2.7 | | |
| | 3.1 | ruby: 3.1 | ruby: 3.x | |
| | 3.2 | ruby: 3.2 | | |

Vous pouvez utiliser une spécification de génération pour installer d'autres composants (par exemple, l'interface AWS CLI, Apache Maven, Apache Ant, Mocha, RSpec ou un composant similaire) lors de la phase de génération `install`. Pour plus d'informations, consultez [Exemple de fichier buildspec](#).

Modes et types de calcul de l'environnement de création

Dans CodeBuild, vous pouvez spécifier l'image de l'environnement de calcul et d'exécution CodeBuild à utiliser pour exécuter vos builds. Le calcul fait référence au moteur informatique (le processeur, la mémoire et le système d'exploitation) qui est géré et maintenu par CodeBuild. Une image d'environnement d'exécution est une image de conteneur qui s'exécute au-dessus de la plateforme de calcul que vous avez choisie et inclut des outils supplémentaires dont votre build pourrait avoir besoin, tels que le AWS CLI.

Rubriques

- [À propos des modes de calcul](#)
- [À propos des types d'environnement](#)

À propos des modes de calcul

CodeBuild propose les modes de calcul suivants :

- EC2
- AWS Lambda

EC2 offre une flexibilité optimisée lors de votre construction et AWS Lambda des vitesses de démarrage optimisées. AWS Lambda prend en charge des builds plus rapides grâce à une latence de démarrage plus faible. AWS Lambda évolue également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Pour de plus amples informations, veuillez consulter [Utilisation du AWS Lambda calcul dans AWS CodeBuild](#).

À propos des types d'environnement

AWS CodeBuild fournit des environnements de construction avec la mémoire, les vCPU et l'espace disque disponibles suivants pour le mode de calcul EC2 :

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | vCPU | Espace disque |
|---------------------------|---------------------------------------|--------------------------------|---------|------|---------------|
| ARM Petit | BUILD_GENERAL1_SMALL | ARM_CONTAINER | 4 Go | 2 | 50 Go |
| ARM Grand | BUILD_GENERAL1_LARGE | ARM_CONTAINER | 16 Go | 8 | 50 Go |
| Linux Small ¹ | BUILD_GENERAL1_SMALL | LINUX_CONTAINER | 3 Go | 2 | 64 Go |
| Linux Medium ¹ | BUILD_GENERAL1_MEDIUM | LINUX_CONTAINER | 7 Go | 4 | 128 Go |
| Linux Large ¹ | BUILD_GENERAL1_LARGE | LINUX_CONTAINER | 15 Go | 8 | 128 Go |

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | vCPU | Espace disque |
|----------------------------|---------------------------------------|--------------------------------|---------|------|---------------|
| Linux XLarge | BUILD_GENERAL1_XLARGE | LINUX_CONTAINER | 70 Go | 36 | 256 Go |
| Linux 2 x Large | BUILD_GENERAL1_2XLARGE | LINUX_CONTAINER | 145 Go | 72 | 824 Go SSD |
| GPU Linux petit | BUILD_GENERAL1_SMALL | LINUX_GPU_CONTAINER | 16 Go | 4 | 220 GO |
| GPU Linux de grande taille | BUILD_GENERAL1_LARGE | LINUX_GPU_CONTAINER | 255 Go | 32 | 50 Go |
| Windows Medium | BUILD_GENERAL1_MEDIUM | WINDOWS_SERVER_2019_CONTAINER | 7 Go | 4 | 128 Go |
| Fenêtres de grande taille | BUILD_GENERAL1_LARGE | WINDOWS_SERVER_2019_CONTAINER | 15 Go | 8 | 128 Go |

¹ La dernière version de ce type d'image est mise en cache. Si vous spécifiez une version plus spécifique, CodeBuild provisionne cette version au lieu de la version mise en cache. Cela peut entraîner des temps de génération plus longs. Par exemple, pour tirer profit de la mise en cache, spécifiez `aws/codebuild/amazonlinux2-x86_64-standard:5.0` à la place d'une version plus précise, telle que `aws/codebuild/amazonlinux2-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild fournit des environnements de construction dotés de la mémoire et de l'espace disque disponibles suivants pour le mode AWS Lambda calcul :

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | Espace disque |
|-------------------|---------------------------------------|--------------------------------|---------|---------------|
| ARM Lambda 1 Go | BUILD_LAMBDA_1GB | ARM_LAMBDA_CONTAINER | 1 Go | 10 Go |
| ARM Lambda 2 Go | BUILD_LAMBDA_2GB | ARM_LAMBDA_CONTAINER | 2 Go | 10 Go |
| ARM Lambda 4 Go | BUILD_LAMBDA_4GB | ARM_LAMBDA_CONTAINER | 4 Go | 10 Go |
| ARM Lambda 8 Go | BUILD_LAMBDA_8GB | ARM_LAMBDA_CONTAINER | 8 Go | 10 Go |
| ARM Lambda 10 Go | BUILD_LAMBDA_10GB | ARM_LAMBDA_CONTAINER | 10 Go | 10 Go |
| Linux Lambda 1 Go | BUILD_LAMBDA_1GB | LINUX_LAMBDA_CONTAINER | 1 Go | 10 Go |
| Linux Lambda 2 Go | BUILD_LAMBDA_2GB | LINUX_LAMBDA_CONTAINER | 2 Go | 10 Go |
| Linux Lambda 4 Go | BUILD_LAMBDA_4GB | LINUX_LAMBDA_CONTAINER | 4 Go | 10 Go |

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | Espace disque |
|--------------------|---------------------------------------|--------------------------------|---------|---------------|
| Linux Lambda 8 Go | BUILD_LAMBDA_8GB | LINUX_LAMBDA_CONTAINER | 8 Go | 10 Go |
| Linux Lambda 10 Go | BUILD_LAMBDA_10GB | LINUX_LAMBDA_CONTAINER | 10 Go | 10 Go |

Lorsque vous utilisez d'autres types d'environnement, il est recommandé d'utiliser une image mise en cache pour réduire les temps de création.

L'espace disque répertorié pour chaque environnement de construction n'est disponible que dans le répertoire spécifié par la variable d'environnement `CODEBUILD_SRC_DIR`.

Pour choisir un type de calcul :

- Dans la CodeBuild console, dans l'assistant de création de projet ou sur la page Modifier le projet de génération, dans Environnement, développez Configuration supplémentaire, puis choisissez l'une des options dans Type de calcul. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).
- Pour le AWS CLI, exécutez la `update-project` commande `create-project` ou en spécifiant la `computeType` valeur de l'environnement objet. Pour plus d'informations, consultez [Création d'un projet de génération \(AWS CLI\)](#) ou [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).
- Pour les AWS SDK, appelez l'équivalent de l'UpdateProject opération `CreateProject` or pour votre langage de programmation cible, en spécifiant l'équivalent de la `computeType` valeur de l'environnement objet. Pour plus d'informations, consultez le [Référence des kits SDK et des outils AWS](#).

Certains types d'environnement et de calcul présentent des limites de disponibilité par région :

- Le type de calcul Linux GPU Small (`LINUX_GPU_CONTAINER`) n'est disponible que dans les régions suivantes :

- USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie Pacifique (Tokyo)
 - Canada (Centre)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
- Le type de calcul Linux GPU Large (LINUX_GPU_CONTAINER) n'est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Séoul)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Le type d'environnement n'ARM_CONTAINER est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Californie du Nord)
 - US West (Oregon)
 - Asie-Pacifique (Hong Kong)
 - Asie-Pacifique (Jakarta)
 - Asie-Pacifique (Hyderabad)

- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Chine (Beijing)
- China (Ningxia)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Espagne)
- Europe (Stockholm)
- Israël (Tel Aviv)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (EAU)
- Amérique du Sud (São Paulo)
- Le type de calcul n'BUILD_GENERAL1_2XLARGE est disponible que dans les régions suivantes :
 - USA Est (Ohio)
 - USA Est (Virginie du Nord)
 - USA Ouest (Californie du Nord)
 - US West (Oregon)
 - Asie-Pacifique (Hyderabad)
 - Asie-Pacifique (Hong Kong)
 - Asie-Pacifique (Jakarta)
 - **Asie-Pacifique (Melbourne)**
- Asie-Pacifique (Mumbai)

- Asie-Pacifique (Séoul)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Canada (Centre)
 - Chine (Beijing)
 - China (Ningxia)
 - Europe (Francfort)
 - Europe (Irlande)
 - Europe (Londres)
 - Europe (Paris)
 - Europe (Espagne)
 - Europe (Stockholm)
 - Europe (Zurich)
 - Israël (Tel Aviv)
 - Moyen-Orient (Bahreïn)
 - Moyen-Orient (EAU)
 - Amérique du Sud (São Paulo)
- Le mode de calcul AWS Lambda (ARM_LAMBDA_CONTAINERetLINUX_LAMBDA_CONTAINER) n'est disponible que dans les régions suivantes :
 - USA Est (Virginie du Nord)
 - USA Est (Ohio)
 - USA Ouest (Oregon)
 - Asie-Pacifique (Mumbai)
 - Asie-Pacifique (Singapour)
 - Asie-Pacifique (Sydney)
 - Asie-Pacifique (Tokyo)
 - Europe (Francfort)
 - Europe (Irlande)
- South America (São Paulo)

Pour le type de calcul BUILD_GENERAL1_2XLARGE, les images Docker jusqu'à 100 Go non compressées sont prises en charge.

Note

Pour les images d'environnement de construction personnalisées, CodeBuild prend en charge les images Docker jusqu'à 50 Go non compressées sous Linux et Windows, quel que soit le type de calcul. Pour vérifier la taille de votre image de génération, utilisez Docker pour exécuter la commande `docker images REPOSITORY:TAG`.

Vous pouvez utiliser Amazon EFS pour accéder à davantage d'espace dans votre conteneur de build. Pour de plus amples informations, veuillez consulter [Exemple d'Amazon Elastic File System pour AWS CodeBuild](#). Si vous souhaitez manipuler l'espace disque du conteneur pendant une génération, la version de la génération doit s'exécuter en mode privilégié.

Note

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

Shells et commandes dans les environnements de génération

Vous fournissez un ensemble de commandes pour qu'AWS CodeBuild les exécute dans un environnement de génération au cours du cycle de vie d'une génération (par exemple, pour installer des dépendances de génération, et tester et compiler votre code source). Vous pouvez spécifier les commandes suivantes de plusieurs façons :

- Créez un fichier de spécification de génération et incluez-le avec votre code source. Dans ce fichier, spécifiez les commandes que vous souhaitez exécuter dans chaque phase du cycle de vie de génération. Pour plus d'informations, consultez l'[Référence de spécification de construction pour CodeBuild](#).
- Utilisez la console CodeBuild pour créer un projet de génération. Dans Insérer des commandes de génération, pour Build commands (Commandes de génération), saisissez les commandes que

vous souhaitez exécuter lors de la phase `build`. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

- Utilisez la console CodeBuild pour modifier les paramètres d'un projet de génération. Dans Insérer des commandes de génération, pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase `build`. Pour plus d'informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).
- Utilisez l'interface AWS CLI ou les kits SDK AWS pour créer un projet de génération ou modifier les paramètres d'un projet de génération. Faites référence au code source qui contient un fichier de spécification de génération avec vos commandes, ou spécifiez une chaîne unique qui inclut le contenu d'un fichier de spécification de génération équivalent. Pour plus d'informations, consultez [Création d'un projet de génération](#) ou [Modification des paramètres d'un projet de génération](#).
- Utilisez l'interface AWS CLI ou les kits AWS SDK pour démarrer une génération, en spécifiant un fichier de spécification de génération ou une chaîne unique qui inclut le contenu d'un fichier de spécification de génération équivalent. Pour de plus amples informations, consultez la description de la valeur `buildspecOverride` dans [Exécution d'une génération](#).

Vous pouvez spécifier n'importe quelle commande Shell Command Language (sh). Dans la version 0.1 de spécification de génération, CodeBuild exécute chaque commande shell dans une instance distincte, dans l'environnement de génération. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par conséquent, par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes (par exemple, pour le changement de répertoire ou la définition de variables d'environnement). Pour contourner ce problème, nous vous recommandons d'utiliser la version 0.2, qui permet de résoudre ce problème. Si vous devez utiliser la version 0.1, nous vous recommandons les approches suivantes :

- Inclure dans votre code source un script shell qui contient les commandes que vous souhaitez exécuter dans une instance unique du shell par défaut. Par exemple, vous pouvez inclure dans votre code source un fichier nommé `my-script.sh` qui contient des commandes comme `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd;`. Ensuite, dans votre fichier de spécification de génération, spécifiez la commande `./my-script.sh`.
- Dans votre fichier de spécification de génération, ou pour le paramètre Build commands (Commandes de génération) pour la phase `build` uniquement, entrez une commande unique incluant toutes les commandes que vous souhaitez exécuter dans une instance unique de l'interpréteur de commandes par défaut (par exemple, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

Si CodeBuild rencontre une erreur, l'erreur peut être plus difficile à résoudre que pour l'exécution d'une commande unique dans sa propre instance du shell par défaut.

Les commandes exécutées dans une image Windows Server Core utilisent l'interpréteur de commandes PowerShell.

Variables d'environnement dans les environnements de génération

AWS CodeBuild fournit plusieurs variables d'environnement que vous pouvez utiliser dans vos commandes de génération :

AWS_DEFAULT_REGION

LaAWS région dans laquelle le build est en cours d'exécution (par exemple,us-east-1). Cette variable d'environnement est utilisée principalement par l'AWS CLI.

AWS_REGION

LaAWS région dans laquelle le build est en cours d'exécution (par exemple,us-east-1). Cette variable d'environnement est utilisée principalement par les kits SDK AWS.

CODEBUILD_BATCH_BUILD_IDENTIFIANT

L'identifiant de la génération dans une génération par lots. Ceci est spécifié dans le batch buildspec. Pour plus d'informations, veuillez consulter [the section called "Référence des spécifications de génération par Batch"](#).

CODEBUILD_BUILD_ARN

Amazon Resource Name (ARN) de laarn:aws:codebuild:*region-ID*:*account-ID*:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE génération.

CODEBUILD_BUILD_ID

L'CodeBuildID du build (par exemple,codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE).

CODEBUILD_BUILD_IMAGE

L'identifiant de l'image deCodeBuild construction (par exemple,aws/codebuild/standard:2.0).

CODEBUILD_BUILD_NUMBER

Numéro de génération pour le projet.

CODEBUILD_BUILD_SUCCEEDING

Si la version actuelle est réussie. Défini sur 0 si la génération échoue, ou sur 1 si la génération aboutit.

CODEBUILD_INITIATOR

Entité qui a commencé la construction. Si CodePipeline a démarré la génération, il s'agit du nom du pipeline (par exemple, `codepipeline/my-demo-pipeline`). Si un utilisateur a lancé la compilation, il s'agit du nom de l'utilisateur (par exemple, `MyUserName`). Si le plug-in Jenkins pour CodeBuild a démarré la génération, il s'agit de la chaîne `CodeBuild-Jenkins-Plugin`.

CODEBUILD_KS_KEY_ID

L'identifiant de laAWS KMS cléCodeBuild utilisée pour crypter l'artefact de sortie de génération (par exemple, `arn:aws:kms:region-ID:account-ID:key/key-ID` ou `alias/key-alias`).

CODEBUILD_LOG_PATH

Le nom du flux deCloudWatch journaux dans Logs pour la génération.

CODEBUILD_PUBLIC_BUILD_URL

URL des résultats de génération pour cette version sur le site Web public des versions. Cette variable n'est définie que si les versions publiques sont activées dans le projet de génération. Pour plus d'informations, veuillez consulter [Projets de build public dansAWS CodeBuild](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

L'identifiant de version du code source d'une version. Le contenu dépend du référentiel de code source :

CodeCommitGitHub, GitHub Enterprise Server et Bitbucket

Cette variable contient l'ID de validation.

CodePipeline

Cette variable contient la révision de la source fournie parCodePipeline.

S'il n'CodePipelineest pas possible de résoudre la révision de la source, par exemple lorsque la source est un compartiment Amazon S3 pour lequel le contrôle de version n'est pas activé, cette variable d'environnement n'est pas définie.

Amazon S3

Cette variable n'est pas définie.

Le cas échéant, la `CODEBUILD_RESOLVED_SOURCE_VERSION` variable n'est disponible qu'après la `DOWNLOAD_SOURCE` phase.

`CODEBUILD_SOURCE_REPO_URL`

URL de l'artefact ou du référentiel de code source en entrée. Pour Amazon S3, cela est `s3://` suivi du nom du compartiment et du chemin d'accès à l'artefact d'entrée. Pour CodeCommit et GitHub, il s'agit de l'URL du clone du référentiel. Si une version provient CodePipeline, cette variable d'environnement peut être vide.

Pour les sources secondaires, la variable d'environnement pour l'URL du référentiel source secondaire est `CODEBUILD_SOURCE_REPO_URL_<sourceIdentifiant>`, où se `<sourceIdentifiant>` trouve l'identifiant de source que vous créez.

`CODEBUILD_SOURCE_VERSION*`

Le format de la valeur dépend du référentiel source.

- Pour Amazon S3, il s'agit de l'ID de version associé à l'artefact d'entrée.
- Pour CodeCommit, c'est l'ID de validation ou le nom de branche associé à la version du code source à générer.
- Pour GitHub GitHub Enterprise Server et Bitbucket, ID de validation, ID de validation, ID de validation, nom de branche ou nom de balise qui correspond à la version du code source à générer.

Note

Pour une version GitHub ou GitHub Enterprise Server déclenchée par un événement de pull request Webhook, c'est le cas `pr/pull-request-number`.

Pour les sources secondaires, la variable d'environnement de la version source secondaire est `CODEBUILD_SOURCE_VERSION_<sourceIdentifiant>`, où se `<sourceIdentifiant>` trouve l'identifiant de source que vous créez. Pour plus d'informations, veuillez consulter [Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#).

`CODEBUILD_SRC_DIR`

Le chemin du répertoire CodeBuild utilisé pour la construction (par exemple, `/tmp/src123456789/src`).

Pour les sources secondaires, la variable d'environnement pour le chemin du répertoire source secondaire est `CODEBUILD_SRC_DIR_<sourceIdentifiant>`, où se `<sourceIdentifiant>`

trouve l'identifiant de source que vous créez. Pour plus d'informations, veuillez consulter [Exemples de projets avec plusieurs sources d'entrée et plusieurs artefacts de sortie](#).

HEURE DE DÉMARRAGE DU CODE

Heure de début de la génération spécifiée sous forme d'horodatage Unix en millisecondes.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

L'ID de compte de l'utilisateur qui a déclenché l'événement webhook.

CODEBUILD_WEBHOOK_BASE_REF

Nom de référence de base de l'événement webhook qui déclenche la version actuelle. Pour une demande d'extraction, c'est ce que l'on appelle la référence de branche.

CODEBUILD_WEBHOOK_EVENT

L'événement webhook qui déclenche la version actuelle.

CODEBUILD_WEBHOOK_MERGE_COMMIT

L'identifiant du commit de fusion utilisé pour la génération. Cette variable est définie lorsqu'une pull request Bitbucket est fusionnée avec la stratégie squash et que la branche pull request est fermée. Dans ce cas, le commit de pull request d'origine n'existe plus. Cette variable d'environnement contient donc l'identifiant du commit de fusion écrasé.

CODEBUILD_WEBHOOK_PREV_COMMIT

L'ID du commit le plus récent avant l'événement push du webhook qui déclenche la version actuelle.

CODEBUILD_WEBHOOK_HEAD_REF

Le nom de référence principal de l'événement webhook qui déclenche la version actuelle. Il peut s'agir d'une référence de branche ou d'une référence de balise.

CODEBUILD_WEBHOOK_TRIGGER

Affiche l'événement webhook qui a déclenché la génération. Cette variable est disponible uniquement pour les builds déclenchés par un webhook. La valeur est analysée à partir de la charge utile envoyée à CodeBuild GitHub Enterprise Server ou Bitbucket. GitHub Le format de la valeur dépend de quel type d'événement a déclenché le build.

- Pour les builds déclenchés par une demande d'extraction, c'est `pr/pull-request-number`.
- Pour les builds déclenchés par la création d'une nouvelle branche ou par l'envoi d'un engagement à une branche, c'est `branch/branch-name`.

- Pour les builds déclenchés par la transmission d'une balise à un référentiel, c'est `tag/tag-name`.

MAISON

Cette variable d'environnement est toujours définie sur `/root`.

Vous pouvez également fournir des environnements de génération avec vos propres variables d'environnement. Pour plus d'informations, consultez les rubriques suivantes :

- [À utiliser CodePipeline avec CodeBuild](#)
- [Création d'un projet de génération](#)
- [Modification des paramètres d'un projet de génération](#)
- [Exécution d'une génération](#)
- [Référence des spécifications de génération](#)

Pour répertorier toutes les variables d'environnement disponibles dans un environnement de génération, vous pouvez exécuter la commande `printenv` (pour les environnements de génération basés sur Linux) ou `"Get-ChildItem Env:"` (les environnements de génération basés sur Windows) pendant une génération. À l'exception de celles précédemment répertoriées, les variables d'environnement qui commencent par `CODEBUILD_` sont réservées à une utilisation interne dans CodeBuild. Elles ne doivent pas être utilisées dans vos commandes de génération.

Important

Nous déconseillons vivement l'utilisation de variables d'environnement pour stocker des valeurs sensibles, en particulier des identifiants de clé AWS d'accès. Les variables d'environnement peuvent être affichées en texte brut à l'aide d'outils comme la console CodeBuild et l'interface AWS CLI.

Nous vous recommandons de stocker les valeurs sensibles dans le magasin de paramètres Amazon EC2 Systems Manager, puis de les récupérer depuis votre buildspec. Pour stocker des valeurs sensibles, consultez la section [Stockage et procédure pas à pas de Systems Manager : création et test d'un paramètre de chaîne \(console\)](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager. Pour les récupérer, consultez le mappage `parameter-store` dans [Syntaxe d'un fichier buildspec](#).

Tâches en arrière-plan dans les environnements de génération

Vous pouvez exécuter des tâches en arrière-plan dans les environnements de génération. Pour ce faire, dans vos spécifications de génération, utilisez la commande `nohup` pour exécuter une commande en tant que tâche en arrière-plan, même si le processus de génération quitte le shell. Utilisez la commande `disown` pour forcer l'arrêt d'une tâche en arrière-plan en cours d'exécution.

Exemples :

- Démarrer un processus en arrière-plan et attendre que celui-ci se termine ultérieurement :

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Démarrer un processus en arrière-plan et ne pas attendre que celui-ci se termine :

```
|  
nohup sleep 30 & disown $!
```

- Démarrer un processus en arrière-plan et y mettre fin ultérieurement :

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
kill $(cat pidfile)
```

Exécutez des builds localement avec AWS CodeBuild agent

Vous pouvez utiliser le `AWS CodeBuild agent` à exécuter `CodeBuilds` appuie sur une machine locale. Des agents sont disponibles pour les plateformes `x86_64` et `ARM`.

Vous pouvez également vous abonner pour recevoir des notifications lorsque de nouvelles versions de l'agent sont publiées.

Prérequis

Avant de commencer, vous devez effectuer les opérations suivantes :

- Installez Git sur votre machine locale.
- Installation et configuration [Docker](#) sur votre machine locale.

Configurer l'image de construction

Vous devez uniquement configurer l'image de génération la première fois que vous exécutez l'agent ou lorsque l'image a été modifiée.

Pour configurer l'image de construction

1. Si vous souhaitez utiliser une image Amazon Linux 2 sélectionnée, vous pouvez l'extraire du CodeBuild référentiel Amazon ECR public à l'adresse https://gallery.ecr.aws/codebuild/amazonlinux2-x86_64-standard avec la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0
```

Sinon, si vous souhaitez utiliser une autre image Linux, effectuez les étapes suivantes :

- a. Clonez le CodeBuild référentiel d'images :

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Accédez au répertoire d'images. Pour cet exemple, utilisez `aws/codebuild/standard:5.0` image :

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Créez l'image. Cela peut prendre plusieurs minutes.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Téléchargez l'agent CodeBuild.

Pour télécharger la version `x86_64` de l'agent, exécutez la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Pour télécharger la version ARM de l'agent, exécutez la commande suivante :

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

- LeCodeBuildl'agent est disponible auprès de<https://gallery.ecr.aws/codebuild/local-builds>.

La signature SHA (Secure Hash Algorithm) pour la version x86_64 de l'agent est la suivante :

```
sha256:fac17c6d6c3cb500f6e7975887de1e41d29a9e70a86d6f49f76a2beacfcf967e
```

La signature SHA pour la version ARM de l'agent est la suivante :

```
sha256:57a5dfda63be50edce13dea16dcd5e73e8d8559029658ba08b793c9a7adc68c7
```

Vous pouvez utiliser le SHA pour identifier la version de l'agent. Pour voir la signature SHA de l'agent, exécutez la commande suivante et recherchez le SHA sousRepoDigests:

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

Exécutez leCodeBuildagent

Pour exécuter leCodeBuildagent

- Accédez au répertoire qui contient la source de votre projet de compilation.
- Téléchargez le[codebuild_build.sh](#)scénario :

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh
$ chmod +x codebuild_build.sh
```

- Exécutez lecodebuild_build.shscript et spécifiez l'image de votre conteneur et le répertoire de sortie.

Pour exécuter une version x86_64, exécutez la commande suivante :

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Pour exécuter une version ARM, exécutez la commande suivante :

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

Remplacer *<container-image>* avec le nom de l'image du conteneur, tel que `aws/codebuild/standard:5.0` ou `public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0`.

Le script lance l'image de génération et exécute la génération sur le projet dans le répertoire actuel. Pour spécifier l'emplacement du projet de construction, ajoutez le -s *<build project directory>* option de la commande de script.

Réception de notifications en cas de nouvelle version de l'agent CodeBuild

Vous pouvez vous abonner aux notifications Amazon SNS afin d'être averti lorsque de nouvelles versions du AWS CodeBuild l'agent est libéré.

Pour vous abonner à CodeBuild notifications aux agents

1. Ouvrez la console Amazon SNS à partir de l'adresse <https://console.aws.amazon.com/sns/v3/home>.
2. Dans la barre de navigation, si ce n'est pas déjà fait, modifiez la AWS Région à USA Est (Virginie du Nord). Vous devez sélectionner cette option AWS Région car les notifications Amazon SNS auxquelles vous êtes abonné sont créées dans cette région.
3. Dans le panneau de navigation, sélectionnez Abonnements.
4. Sélectionnez Créer un abonnement.
5. Dans Créer un abonnement, procédez comme suit :

- a. Pour ARN de la rubrique, utilisez l'Amazon Resource Name (ARN) suivant :

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. Pour Protocol (Protocole), choisissez Email (E-mail) ou SMS.
- c. Pour Endpoint (Point de terminaison), choisissez où recevoir les notifications (e-mail ou SMS). Saisissez une adresse e-mail, une adresse ou un numéro de téléphone, y compris l'indicatif régional.
- d. Choisissez Create subscription (Créer un abonnement).

- e. Choisissez Courrier électronique pour recevoir un e-mail vous demandant de confirmer votre abonnement. Suivez les instructions dans l'e-mail pour finaliser votre abonnement.

Si vous ne souhaitez plus recevoir ces notifications, exécutez la procédure suivante pour annuler votre abonnement.

Pour annuler votre abonnement aux notifications relatives à l'agent CodeBuild :

1. Ouvrez la console Amazon SNS à partir de l'adresse <https://console.aws.amazon.com/sns/v3/home>.
2. Dans le panneau de navigation, choisissez Abonnements.
3. Sélectionnez l'abonnement, puis sous Actions, choisissez Delete subscriptions (Supprimer des abonnements). Lorsque vous êtes invité à confirmer, choisissez Delete.

Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud

En règle générale, AWS CodeBuild ne peut pas accéder aux ressources d'un VPC. Pour activer l'accès, vous devez fournir des informations de configuration supplémentaires spécifiques au VPC dans le cadre de votre configuration de projet CodeBuild. Cela inclut l'ID de VPC, les ID du sous-réseau VPC et les ID du groupe de sécurité du VPC. Les builds compatibles VPC peuvent ensuite accéder aux ressources à l'intérieur de votre VPC. Pour plus d'informations sur la configuration d'un VPC dans Amazon VPC, consultez le [Guide de l'utilisateur d'Amazon VPC](#).

Rubriques

- [Cas d'utilisation](#)
- [Autoriser l'accès à Amazon VPC dans votre CodeBuild projets](#)
- [Bonnes pratiques pour les VPC](#)
- [Dépannage de la configuration de votre VPC](#)
- [Limites des VPC](#)
- [Utilisation de points de terminaison de VPC](#)
- [Modèle VPC AWS CloudFormation](#)
- [Utilisation d'AWS CodeBuild avec un serveur proxy](#)

Cas d'utilisation

La connectivité VPC à partir de générations AWS CodeBuild rend cela possible.

- Exécutez des tests d'intégration à partir de votre build par rapport aux données d'une base de données Amazon RDS isolée sur un sous-réseau privé.
- Rechercher des données dans un AmazonElastiCachecluster directement à partir des tests.
- Interagissez avec des services Web internes hébergés sur Amazon EC2, Amazon ECS ou des services qui utilisent Elastic Load Balancing interne.
- Récupérez des dépendances de référentiels d'artefact internes auto-hébergés, comme PyPI pour Python, Maven pour Java et npm pour Node.js.
- Accédez aux objets d'un compartiment S3 configuré pour autoriser l'accès via un point de terminaison Amazon VPC uniquement.

- Recherchez des services web externes qui nécessitent des adresses IP fixes par le biais de l'adresse IP Elastic de la passerelle NAT ou d'une instance NAT associée à votre ou vos sous-réseaux.

Vos générations peuvent accéder à n'importe quelle ressource qui est hébergée dans votre VPC.

Autoriser l'accès à Amazon VPC dans votre CodeBuild projets

Incluez ces paramètres dans la configuration de votre VPC :

- Pour `For VPC ID` (ID de VPC), choisissez l'ID de VPC utilisé par CodeBuild.
- Pour `Subnets` (Sous-réseaux), choisissez un sous-réseau privé avec traduction NAT qui comporte ou possède des routes vers les ressources CodeBuild utilisées.
- Pour `Security Groups` (Groupes de sécurité), choisissez les groupes de sécurité utilisés par CodeBuild pour autoriser l'accès aux ressources des VPC.

Pour utiliser la console afin de créer un projet de génération, consultez [Création d'un projet de génération \(console\)](#). Lorsque vous créez ou modifiez votre projet CodeBuild, dans VPC, choisissez votre ID de VPC, des sous-réseaux et des groupes de sécurité.

Pour utiliser l'AWS CLI afin de créer un projet de génération, consultez [Création d'un projet de génération \(AWS CLI\)](#). Si vous utilisez l'AWS CLI avec CodeBuild, le rôle de service utilisé par CodeBuild pour interagir avec les services au nom de l'utilisateur IAM doit être associé à une stratégie. Pour plus d'informations, consultez [Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC](#).

Le `Configuration VPC` l'objet doit inclure votre `Identifiant VPCId`, `securityGroupIds`, et `sous-réseaux`.

- `vpcId` : obligatoire. ID de VPC utilisé par CodeBuild. Exécutez cette commande pour obtenir la liste de tous les identifiants Amazon VPC de votre région :

```
aws ec2 describe-vpcs
```

- `subnets` : obligatoire. ID de sous-réseau qui comprennent les ressources utilisées par CodeBuild. Exécutez cette commande pour obtenir ces identifiants :

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Remplacez `us-east-1` par votre région.

- ***securityGroupIds*** : obligatoire. ID de groupe de sécurité utilisés par CodeBuild pour permettre l'accès aux ressources du VPC. Exécutez cette commande pour obtenir ces identifiants :

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Remplacez `us-east-1` par votre région.

Bonnes pratiques pour les VPC

Utilisez cette liste de contrôle lors de la configuration d'un VPC à utiliser avec CodeBuild.

- Configurez votre VPC avec des sous-réseaux publics et privés, ainsi qu'une passerelle NAT. La passerelle NAT doit résider dans un sous-réseau public. Pour plus d'informations, consultez [VPC avec des sous-réseaux publics et privés \(NAT\)](#) dans le Amazon VPC Guide de l'utilisateur.

⚠ Important

Vous avez besoin d'une passerelle NAT ou d'une instance NAT pour utiliser CodeBuild avec votre VPC afin que CodeBuild peut atteindre des points de terminaison publics (par exemple, pour exécuter des commandes CLI lors de l'exécution de versions). Vous ne pouvez pas utiliser la passerelle Internet à la place d'une passerelle NAT ou d'une instance NAT car CodeBuild ne prend pas en charge l'attribution d'adresses IP élastiques aux interfaces réseau qu'il crée, et l'attribution automatique d'une adresse IP publique n'est pas prise en charge par Amazon EC2 pour les interfaces réseau créées en dehors des lancements d'instances Amazon EC2.

- Incluez plusieurs zones de disponibilité avec votre VPC.

- Assurez-vous que vos groupes de sécurité n'ont pas de trafic entrant (d'entrée) autorisé sur vos générations. CodeBuild n'a pas d'exigences spécifiques pour le trafic sortant, mais vous devez autoriser l'accès à toutes les ressources Internet requises pour votre build, telles que GitHub ou Amazon S3.

Pour de plus amples informations, veuillez consulter [Règles des groupes de sécurité](#) dans le Guide de l'utilisateur Amazon VPC.

- Configurez des sous-réseaux distincts pour vos générations.
- Lorsque vous configurez vos projets CodeBuild pour accéder à votre VPC, choisissez des sous-réseaux privés uniquement.

Pour plus d'informations sur la configuration d'un VPC dans Amazon VPC, consultez le [Guide de l'utilisateur d'Amazon VPC](#).

Pour en savoir plus sur l'utilisation d'AWS CloudFormation pour configurer un VPC de manière à utiliser la fonctionnalité VPC CodeBuild, consultez [Modèle VPC AWS CloudFormation](#).

Dépannage de la configuration de votre VPC

Utilisez les informations affichées dans le message d'erreur pour vous aider à identifier, à diagnostiquer et à résoudre les problèmes.

Voici quelques instructions pour vous aider lors du dépannage d'une erreur VPC CodeBuild courante : Build does not have internet connectivity. Please check subnet network configuration.

1. [Assurez-vous que votre passerelle Internet est attachée au VPC.](#)
2. [Assurez-vous que la table de routage de votre sous-réseau public pointe vers la passerelle Internet.](#)
3. [Assurez-vous que vos ACL réseau autorisent la circulation du trafic.](#)
4. [Assurez-vous que vos groupes de sécurité autorisent la circulation du trafic.](#)
5. [Dépannez votre passerelle NAT.](#)
6. [Assurez-vous que la table de routage des sous-réseaux privés pointe vers la passerelle NAT.](#)
7. Assurez-vous que le rôle de service utilisé par CodeBuild pour interagir avec les services au nom de l'utilisateur IAM dispose des autorisations dans [cette stratégie](#). Pour plus d'informations, veuillez consulter [Création d'un rôle CodeBuild de service](#).

Si CodeBuild ne dispose pas de toutes les autorisations requises, vous pouvez recevoir l'erreur `Unexpected EC2 error: UnauthorizedOperation`. Cette erreur peut se produire si CodeBuild ne dispose pas des autorisations Amazon EC2 requises pour travailler avec un VPC.

Limites des VPC

- La connectivité de VPC à partir d'CodeBuild n'est pas prise en charge dans Windows.
- Connectivité VPC depuis CodeBuild n'est pas pris en charge pour les VPC partagés.
- Si vous configurez un VPC, [mise en cache locale](#) n'est pas pris en charge. À compter du 28/02/2022, la génération de votre VPC prendra plus de temps car une nouvelle instance Amazon EC2 sera utilisée pour chaque build.

Utilisation de points de terminaison de VPC

Vous pouvez améliorer la sécurité de vos générations en configurant AWS CodeBuild pour qu'il utilise un point de terminaison de VPC d'interface. Les points de terminaison de l'interface sont alimentés par PrivateLink, une technologie que vous pouvez utiliser pour accéder en privé à Amazon EC2 et CodeBuild en utilisant des adresses IP privées. PrivateLink restreint tout le trafic réseau entre vos instances gérées, CodeBuild, et Amazon EC2 sur le réseau Amazon. (Les instances gérées n'ont pas accès à Internet). De même, vous n'avez pas besoin d'une passerelle Internet, d'un périphérique NAT ni d'une passerelle privée virtuelle. Il n'est pas obligatoire de configurer PrivateLink, mais c'est recommandé. Pour plus d'informations sur PrivateLink et points de terminaison VPC, voir [Qu'est-ce que AWS PrivateLink?](#).

Avant de créer des points de terminaison de VPC

Avant de configurer les points de terminaison d'un VPC pour AWS CodeBuild, tenez compte des restrictions et limitations suivantes.

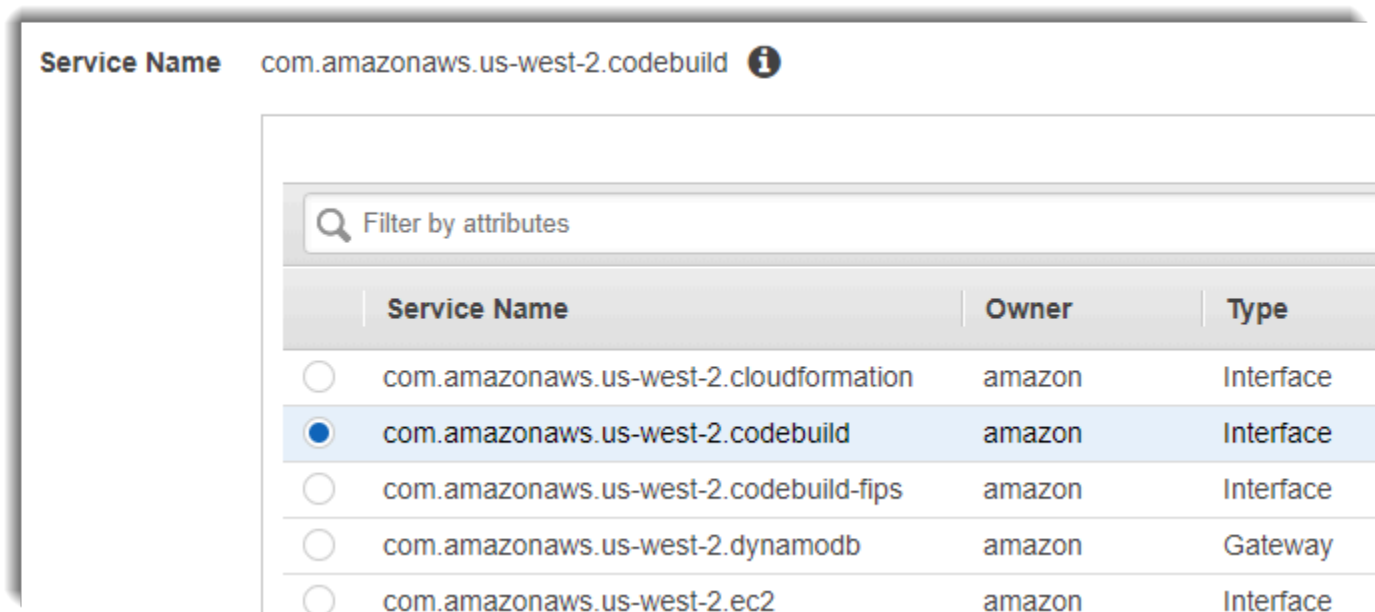
Note

Utilisez un [Passerelle NAT](#) si vous souhaitez utiliser CodeBuild avec AWS services qui ne prennent pas en charge Amazon VPC PrivateLink connexions.

- Les points de terminaison VPC prennent en charge le DNS fourni par Amazon via Amazon Route 53 uniquement. Si vous souhaitez utiliser votre propre DNS, vous pouvez utiliser le transfert DNS conditionnel. Pour plus d'informations, consultez [Jeux d'options DHCP](#) dans le Guide de l'utilisateur Amazon VPC.
- Les points de terminaison de VPC ne prennent pas en charge les demandes inter-régionales pour le moment. Assurez-vous de créer votre point de terminaison dans la même AWS Région que tous les compartiments S3 qui stockent les entrées et sorties de votre build. Vous pouvez utiliser la console Amazon S3 ou `get-bucket-location` commande pour trouver l'emplacement de votre compartiment. Utilisez un point de terminaison Amazon S3 spécifique à une région pour accéder à votre compartiment (par exemple, `<bucket-name>.s3-us-west-2.amazonaws.com`). Pour plus d'informations sur les points de terminaison spécifiques à une région pour Amazon S3, consultez [Service de stockage Amazon Simple](#) dans la Référence générale d'Amazon Web Services. Si vous utilisez le AWS CLI pour envoyer des demandes à Amazon S3, définissez votre région par défaut sur la même région que celle dans laquelle votre compartiment a été créé, ou utilisez `--region` paramètre dans vos demandes.

Création des points de terminaison d'un VPC pour CodeBuild

Suivez les instructions indiquées dans [Création d'un point de terminaison d'interface](#) pour créer le point de terminaison `com.amazonaws.us-west-2.codebuild`. Ceci est un point de terminaison VPC pour AWS CodeBuild.



The screenshot shows the AWS Management Console interface for the region us-west-2. The 'Service Name' is 'com.amazonaws.us-west-2.codebuild'. Below this, there is a search bar labeled 'Filter by attributes'. A table lists several VPC endpoints, with 'com.amazonaws.us-west-2.codebuild' highlighted in blue. The table has columns for 'Service Name', 'Owner', and 'Type'.

| | Service Name | Owner | Type |
|----------------------------------|--|--------|-----------|
| <input type="radio"/> | com.amazonaws.us-west-2.cloudformation | amazon | Interface |
| <input checked="" type="radio"/> | com.amazonaws.us-west-2.codebuild | amazon | Interface |
| <input type="radio"/> | com.amazonaws.us-west-2.codebuild-fips | amazon | Interface |
| <input type="radio"/> | com.amazonaws.us-west-2.dynamodb | amazon | Gateway |
| <input type="radio"/> | com.amazonaws.us-west-2.ec2 | amazon | Interface |

region représente l'identifiant de région d'une région AWS prise en charge par CodeBuild, telle que `us-east-2` pour la région USA Est (Ohio). Pour une liste des produits pris en charge AWS Régions, voir [CodeBuild](#) dans le AWS Référence générale. Le point de terminaison est prérempli avec la région que vous avez spécifiée lorsque vous vous êtes connecté à AWS. Si vous changez de région, le point de terminaison du VPC est mis à jour en conséquence.

Création d'une stratégie de point de terminaison de VPC pour CodeBuild

Vous pouvez créer une politique pour les points de terminaison Amazon VPC pour AWS CodeBuild dans lequel vous pouvez spécifier :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources qui peuvent avoir des actions exécutées sur elles.

L'exemple de stratégie suivant spécifie que tous les mandataires peuvent uniquement lancer et afficher des générations pour le projet `project-name`.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Pour plus d'informations, consultez [Contrôle de l'accès aux services avec points de terminaison d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Modèle VPC AWS CloudFormation

AWS CloudFormation vous permet de créer et de mettre en service AWS déploiements d'infrastructure de manière prévisible et répétée à l'aide de fichiers de modèle pour créer et supprimer un ensemble de ressources en tant qu'unité unique (a)empiler). Pour plus d'informations, veuillez consulter le [Guide de l'utilisateur AWS CloudFormation](#).

Voici un modèle AWS CloudFormation YAML permettant de configurer un VPC pour utiliser AWS CodeBuild. également disponible dans la [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::VPCElasticNetworkInterfaceAttachment
```

```
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
```

Utilisation d'AWS CodeBuild avec un serveur proxy

Vous pouvez utiliser AWS CodeBuild un serveur proxy pour réguler le trafic HTTP et HTTPS depuis et vers Internet. Pour exécuter CodeBuild avec un serveur proxy, vous installez un serveur proxy dans un sous-réseau public et CodeBuild dans un sous-réseau privé dans un VPC.

Il existe deux cas d'utilisation principaux pour exécuter CodeBuild sur un serveur proxy :

- Il élimine l'utilisation d'une passerelle NAT ou d'une instance NAT dans votre VPC.
- Il vous permet de spécifier les URL auxquelles les instances du serveur proxy peuvent accéder et les URL auxquelles le serveur proxy refuse l'accès.

Vous pouvez utiliser CodeBuild avec deux types de serveur proxy. Dans les deux cas, le serveur proxy s'exécute dans un sous-réseau public et CodeBuild s'exécute dans un sous-réseau privé.

- Proxy explicite : Si vous utilisez un serveur proxy explicite, vous devez configurer `NO_PROXY`, `HTTP_PROXY`, et `HTTPS_PROXY` variables d'environnement dans CodeBuild au niveau du projet. Pour plus d'informations, consultez [Modification des paramètres d'un projet de génération dans AWS CodeBuild](#) et [Création d'un projet de génération dans AWS CodeBuild](#).
- Proxy Transparent : Si vous utilisez un serveur proxy transparent, aucune configuration spéciale n'est nécessaire.

Rubriques

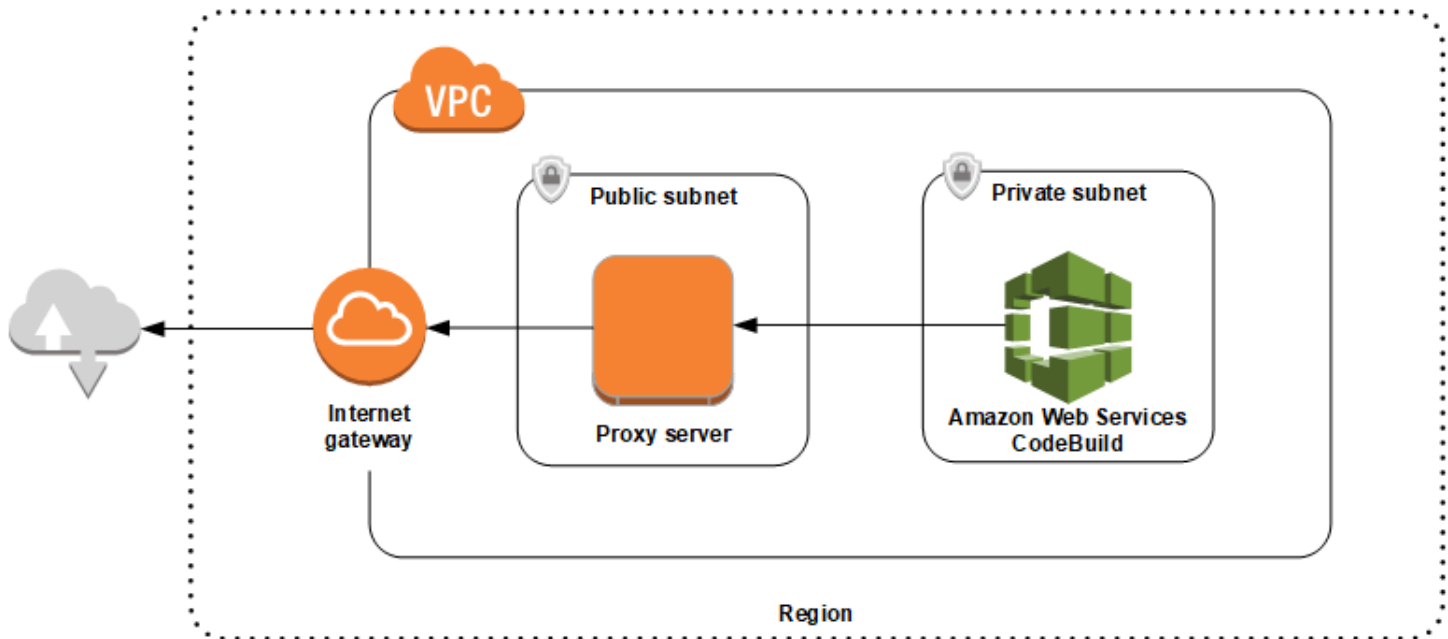
- [Composants requis pour exécuter CodeBuild sur un serveur proxy](#)
- [Exécution de CodeBuild sur un serveur proxy explicite](#)
- [Exécution de CodeBuild sur un serveur proxy transparent](#)
- [Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy](#)

Composants requis pour exécuter CodeBuild sur un serveur proxy

Ces composants doivent exécuter AWS CodeBuild dans un serveur proxy transparent ou explicite :

- Un VPC.
- Un sous-réseau public dans votre VPC pour le serveur proxy
- Un sous-réseau privé dans votre VPC pour CodeBuild.
- Une passerelle Internet qui permet la communication entre le VPC et Internet.

Le schéma suivant illustre la manière dont les deux composants interagissent.



Configuration d'un VPC, de sous-réseaux et d'une passerelle réseau

Les étapes suivantes sont requises pour exécuter AWS CodeBuild dans un serveur proxy transparent ou explicite.

1. Créez un VPC. Pour plus d'informations, consultez [Création d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.
2. Créez deux sous-réseaux dans votre VPC. L'un des deux est un sous-réseau public, nommé `Public Subnet`, dans lequel le serveur proxy s'exécute. L'autre est un sous-réseau privé, nommé `Private Subnet` dans lequel CodeBuild s'exécute.

Pour de plus amples informations, veuillez consulter [Création d'un sous-réseau dans votre VPC](#).

3. Créez et attachez une passerelle Internet à votre VPC. Pour de plus amples informations, veuillez consulter [Création et attachement d'une passerelle Internet](#).
4. Ajoutez une règle à la table de routage par défaut qui achemine le trafic sortant à partir du VPC (0.0.0.0/0) vers la passerelle Internet. Pour de plus amples informations, veuillez consulter [Ajout et retrait de routes d'une table de routage](#).
5. Ajoutez une règle au groupe de sécurité par défaut de votre VPC qui autorise le trafic SSH entrant (TCP 22) à partir de votre VPC (0.0.0.0/0).
6. Suivez les instructions de la section [Lancement d'une instance à l'aide de l'assistant de lancement d'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour lancer une instance Amazon Linux. Lorsque vous exécutez l'assistant, choisissez les options suivantes :

- Dans Choisir un type d'instance, choisissez une Amazon Machine Image (AMI) Amazon Linux.
- Dans Subnet (Sous-réseau), choisissez le sous-réseau public que vous avez créé précédemment dans cette rubrique. Si vous avez utilisé le nom suggéré, il s'agit de Public Subnet.
- Pour Auto-assign Public IP (Attribuer automatiquement) l'adresse IP publique, sélectionnez (Enable) Activer.
- Sur la page Configure Security Group (Configurer un groupe de sécurité), pour Assign a security group (Affecter un groupe de sécurité), choisissez Select an existing security group (Sélectionner un groupe de sécurité existant). Ensuite, choisissez le groupe de sécurité par défaut.
- Après avoir choisi Launch (Lancer), choisissez une paire de clés existante ou créez-en une.

Choisissez les paramètres par défaut pour toutes les autres options.

7. Une fois que votre instance EC2 est en cours d'exécution, désactivez les vérifications de source/destination. Pour obtenir des informations, veuillez consulter [Désactivation des vérifications de source/destination](#) dans le Guide de l'utilisateur Amazon VPC.
8. Créez une table de routage dans votre VPC. Ajoutez une règle à la table de routage qui achemine le trafic destiné à Internet vers votre serveur proxy. Associez cette table de routage à votre sous-réseau privé. Cette action est obligatoire pour que les demandes sortantes à partir des instances de votre sous-réseau privé, où CodeBuild s'exécute, soient toujours acheminées via le serveur proxy.

Installation et configuration d'un serveur proxy

Il existe de nombreux serveurs proxy parmi lesquels choisir. Un serveur proxy open source, Squid, est utilisé ici pour expliquer la façon dont AWS CodeBuild s'exécute dans un serveur proxy. Vous pouvez appliquer les mêmes concepts à d'autres serveurs proxy.

Pour installer Squid, utilisez un référentiel yum en exécutant les commandes suivantes :

```
sudo yum update -y
sudo yum install -y squid
```

Après avoir installé Squid, modifiez son fichier `squid.conf` en suivant les instructions plus loin dans cette rubrique.

Configuration de Squid pour le trafic HTTPS

Pour HTTPS, le trafic HTTP est encapsulé dans une connexion à protocole TLS (Transport Layer Security, Sécurité de la couche transport). Squid utilise une fonction appelée [SslPeekAndSplice](#) pour extraire l'indication de nom de serveur (SNI) à partir de l'initiation TLS qui contient l'hôte internet demandé. Cette opération est requise pour que Squid ne soit pas obligé de déchiffrer le trafic HTTPS. Pour activer SslPeekAndSplice, Squid a besoin d'un certificat. Créez ce certificat à l'aide d'OpenSSL :

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

Pour HTTP, Squid ne nécessite aucune configuration. À partir de tous les messages de demande HTTP/1.1, il peut extraire le champ d'en-tête de l'hôte, qui spécifie l'hôte Internet demandé.

Exécution de CodeBuild sur un serveur proxy explicite

Rubriques

- [Configuration de Squid en tant que serveur proxy explicite](#)
- [Création d'un projet CodeBuild](#)
- [Exemple de fichier squid.conf de serveur proxy explicite](#)

Pour exécuter AWS CodeBuild sur un serveur proxy explicite, vous devez configurer ce dernier pour autoriser ou refuser le trafic vers et depuis des sites externes, puis configurer les variables d'environnement HTTPS_PROXY et HTTP_PROXY.

Configuration de Squid en tant que serveur proxy explicite

Pour configurer le serveur proxy Squid en tant que serveur explicite, vous devez apporter les modifications suivantes au fichier `/etc/squid/squid.conf` :

- Supprimez les règles de liste de contrôle d'accès (ACL) par défaut suivantes :

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Ajoutez les règles suivantes à la place des règles ACL par défaut que vous avez supprimées. La première ligne autorise les demandes à partir de votre VPC. Les deux lignes suivantes accordent à votre serveur proxy l'accès aux URL de destination qui peuvent être utilisées par AWS CodeBuild. Modifiez l'expression régulière de la dernière ligne pour spécifier des compartiments S3 ou un référentiel CodeCommit dans une AWS Région . Par Exemple:

- Si votre source est Amazon S3, utilisez la commande `acl download_src dstdomain_regex .*s3\.us-west-1\.amazonaws\.com` pour accorder l'accès aux compartiments S3 dans le `us-west-1` Région .
- Si votre source est AWS CodeCommit, utilisez `git-codecommit.<your-region>\.amazonaws\.com` pour ajouter une AWS Région à une liste d'autorisation.

```
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 dstdomain_regex .*\.amazonaws\.com #Allows to download source from Amazon S3 or CodeCommit
```

- Remplacez `http_access allow localnet` par ce qui suit :

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Si vous souhaitez que votre build charge des journaux et des artefacts, effectuez l'une des actions suivantes :

1. Avant l'instruction `http_access deny all`, insérez les instructions suivantes. Ils permettent à CodeBuild d'accéder à CloudWatch et Amazon S3. L'accès à CloudWatch est requis pour que CodeBuild puisse créer des journaux CloudWatch. L'accès à Amazon S3 est obligatoire pour le chargement des artefacts et la mise en cache 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
```

- Une fois que vous avez enregistré `squid.conf`, exécutez la commande suivante :

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Ajoutez proxy à votre fichier `buildspec`. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
      - command
```

Note

Si vous recevez une erreur de délai d'attente `RequestError`, consultez [RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy](#).

Pour plus d'informations, consultez [Exemple de fichier squid.conf de serveur proxy explicite](#) plus loin dans cette rubrique.

Création d'un projet CodeBuild

Pour exécuter AWS CodeBuild avec votre serveur proxy explicite, définissez ses variables d'environnement `HTTPS_PROXY` et `HTTP_PROXY` avec l'adresse IP privée de l'instance EC2 que

vous avez créée pour votre serveur proxy et le port 3128 au niveau du projet. L'adresse IP privée se présente ainsi : `http://your-ec2-private-ip-address:3128`. Pour plus d'informations, consultez [Création d'un projet de génération dans AWS CodeBuild](#) et [Modification des paramètres d'un projet de génération dans AWS CodeBuild](#).

Utilisez la commande suivante pour afficher le journal d'accès proxy Squid :

```
sudo tail -f /var/log/squid/access.log
```

Exemple de fichier **squid.conf** de serveur proxy explicite

L'exemple suivant présente un fichier `squid.conf` configuré pour un serveur proxy explicite.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
# add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
```



```
http_access deny manager
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
# Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
# And finally deny all other access to this proxy
http_access deny all
# Squid normally listens to port 3128
http_port 3128
# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

Exécution de CodeBuild sur un serveur proxy transparent

Pour exécuter AWS CodeBuild dans un serveur proxy transparent, vous devez configurer le serveur proxy avec un accès aux sites Web et aux domaines avec lesquels il interagit.

Configuration de Squid en tant que serveur proxy transparent

Pour configurer un serveur proxy en tant que serveur transparent, vous devez lui accorder l'accès aux domaines et aux sites Web auxquels vous souhaitez qu'il accède. Pour exécuter AWS CodeBuild avec un serveur proxy transparent, vous devez lui accorder l'accès à `amazonaws.com`. Vous devez également accorder l'accès aux autres sites Web utilisés par CodeBuild. Ces derniers varient en fonction de la façon dont vous créez vos projets CodeBuild. Par exemple, il peut s'agir de sites Web pour des référentiels tels que GitHub, Bitbucket, Yum et Maven. Pour accorder à Squid l'accès à des domaines et à des sites Web spécifiques, utilisez une commande similaire à la suivante pour mettre à jour le fichier `squid.conf`. Cet exemple de commande accorde l'accès à `amazonaws.com`, `github.com` et `bitbucket.com`. Vous pouvez modifier cet exemple pour donner accès à d'autres sites Web.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

```
http_access deny all
EOF
```

Les demandes entrantes à partir des instances du sous-réseau privé doivent être redirigées vers les ports Squid. Squid écoute le trafic HTTP sur le port 3129 (au lieu de 80) et le trafic HTTPS sur le port 3130 (au lieu de 443). Utilisez la commande iptables pour acheminer le trafic :

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

Création d'un projet CodeBuild

Après avoir configuré votre serveur proxy, vous pouvez l'utiliser avec AWS CodeBuild dans un sous-réseau privé sans davantage de configuration. Chaque demande HTTP et HTTPS passe par le serveur proxy public. Utilisez la commande suivante pour afficher le journal d'accès proxy Squid :

```
sudo tail -f /var/log/squid/access.log
```

Exécution d'un gestionnaire de package et d'autres outils sur un serveur proxy

Pour exécuter un outil tel qu'un gestionnaire de package sur un serveur proxy

1. Pour inclure l'outil dans la liste d'autorisation de votre serveur proxy, ajoutez des instructions dans le fichier `squid.conf`.
2. Ajoutez une ligne à votre fichier `buildspec` qui pointe vers le point de terminaison privé de votre serveur proxy.

Les exemples suivants montrent comment procéder pour `apt-get`, `curl` et `maven`. Si vous utilisez un outil différent, les mêmes principes s'appliquent. Ajoutez-le à une liste d'autorisation dans `squid.conf` et ajoutez une commande à votre fichier `buildspec` pour faire connaître à CodeBuild le point de terminaison de votre serveur proxy.

Exécution de **apt-get** sur un serveur proxy

1. Ajoutez les instructions suivantes à votre fichier `squid.conf` pour ajouter `apt-get` à une liste d'autorisation sur votre serveur proxy. Les trois premières lignes permettent `apt-get` à exécuter dans l'environnement de construction.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Ajoutez l'instruction suivante dans votre fichier `buildspec` afin que les commandes `apt-get` recherchent la configuration proxy dans `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

Exécution de **curl** sur un serveur proxy

1. Insérez ce qui suit dans votre fichier `squid.conf` pour ajouter `curl` à une liste d'autorisation dans votre environnement de génération.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a webiste. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Ajoutez l'instruction suivante dans votre fichier `buildspec` de sorte que `curl` utilise le serveur proxy privé pour accéder au site Web que vous avez ajouté à `squid.conf`. Dans cet exemple, le site Web est `google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

Exécution de **maven** sur un serveur proxy

1. Insérez ce qui suit dans votre fichier `squid.conf` pour ajouter maven à une liste d'autorisation dans votre environnement de génération.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
  build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
  build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Ajoutez l'instruction suivante à votre fichier `buildspec`.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

Utilisation des projets de génération et des générations dans AWS CodeBuild

Pour commencer, suivez les étapes de [Création d'un projet de génération](#), puis suivez les étapes de [Exécution d'une génération](#). Pour plus d'informations sur les projets de génération et les générations, consultez les rubriques suivantes.

Rubriques

- [Utilisation des projets de génération](#)
- [Gestion des générations dans AWS CodeBuild](#)

Utilisation des projets de génération

Un projet de compilation inclut des informations sur la façon d'exécuter une compilation, notamment sur l'emplacement du code source, sur l'environnement de compilation à utiliser, sur les commandes de compilation à exécuter et sur l'emplacement où stocker le résultat de la compilation.

Vous pouvez effectuer ces tâches lorsque vous gérez des projets de génération :

Rubriques

- [Création d'un projet de génération dans AWS CodeBuild](#)
- [Création d'une règle de notification](#)
- [Affichage d'une liste de noms de projet de génération dans AWS CodeBuild](#)
- [Affichage des détails d'un projet de génération dans AWS CodeBuild](#)
- [Mise en cache de génération dans AWS CodeBuild](#)
- [Intégrez des déclencheurs dans AWS CodeBuild](#)
- [Utilisation de webhooks avec AWS CodeBuild](#)
- [Modification des paramètres d'un projet de génération dans AWS CodeBuild](#)
- [Suppression d'un projet de génération dans AWS CodeBuild](#)
- [Utilisation de projets partagés](#)
- [Balisage de projets dans AWS CodeBuild](#)
- [Batch s'intègre AWS CodeBuild](#)
- [GitHubAction Runner dans AWS CodeBuild](#)

- [Projets de build public dans AWS CodeBuild](#)

Création d'un projet de génération dans AWS CodeBuild

Vous pouvez utiliser la console AWS CodeBuild, l'AWS CLI ou les kits SDK AWS pour créer un projet de génération.

Prérequis

Avant de créer un projet de construction, répondez aux questions de [Planification d'une génération](#).

Rubriques

- [Création d'un projet de génération \(console\)](#)
- [Création d'un projet de génération \(AWS CLI\)](#)
- [Création d'un projet de génération \(kits SDK AWS\)](#)
- [Création d'un projet de génération \(AWS CloudFormation\)](#)

Création d'un projet de génération (console)

Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).

Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.

Choisissez Créer un projet de génération.

Renseignez les sections suivantes. Une fois terminé, choisissez Créer un projet de construction au bas de la page.

Rubriques :

- [Configuration du projet](#)
- [Source](#)
- [Environnement](#)
- [Spécifications de construction](#)
- [Configuration par lots](#)

- [Artefacts](#)
- [Journaux](#)

Configuration du projet

Nom du projet

Entrez un nom pour ce projet de construction. Les noms de projet de génération doivent être uniques dans chaque compte AWS.

Description

Entrez une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

Construire un badge

(Facultatif) Sélectionnez Activer le badge de construction pour rendre l'état de construction de votre projet visible et intégrable. Pour plus d'informations, consultez [Exemple de badges de génération](#).

Note

Le badge de construction ne s'applique pas si votre fournisseur source est Amazon S3.

Activer la limite de génération simultanée

(Facultatif) Si vous souhaitez limiter le nombre de builds simultanés pour ce projet, effectuez les opérations suivantes :

1. Sélectionnez Restreindre le nombre de versions simultanées que ce projet peut démarrer.
2. Dans Limite de génération simultanée, entrez le nombre maximum de versions simultanées autorisées pour ce projet. Cette limite ne peut pas être supérieure à la limite de création simultanée définie pour le compte. Si vous essayez de saisir un nombre supérieur à la limite du compte, un message d'erreur s'affiche.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Informations supplémentaires

(Facultatif) Pour Balises, saisissez le nom et la valeur des balises que vous souhaitez que les services AWS de prise en charge utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.

Source

Fournisseur de source

Choisissez le type de fournisseur de code source. Utilisez les listes suivantes pour effectuer des sélections adaptées à votre fournisseur de source :

Note

CodeBuild ne prend pas en charge Bitbucket Server.

Amazon S3

Compartiment

Choisissez le nom du compartiment d'entrée contenant le code source.

Clé d'objet S3 ou dossier S3

Entrez le nom du fichier ZIP ou le chemin d'accès au dossier contenant le code source. Entrez une barre oblique (/) pour tout télécharger dans le compartiment S3.

Version de la source

Entrez l'ID de version de l'objet qui représente la version de votre fichier d'entrée. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

CodeCommit

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Type de référence

Choisissez Branch, Git tag ou Commit ID pour spécifier la version de votre code source. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez de créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Bitbucket

Référentiel

Choisissez Connect using OAuth ou Connect with a Bitbucket app password, puis suivez les instructions pour vous connecter (ou vous reconnecter) à Bitbucket.

Choisissez un dépôt public ou un dépôt dans votre compte.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, veuillez consulter la page [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que

811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le name paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

Pour l'URL cible, entrez la valeur à utiliser pour le url paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code

source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [Événements du webhook Bitbucket](#).

GitHub

Référentiel

Choisissez Connect using OAuth ou Connect with a GitHub personal access token, puis suivez les instructions pour vous connecter (ou vous reconnecter) GitHub et autoriser l'accès à AWS CodeBuild

Choisissez un dépôt public ou un dépôt dans votre compte.

Version de la source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, veuillez consulter la page [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte d'état, entrez la valeur à utiliser pour le `context` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour l'URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitHub Enterprise Server

GitHub Jeton d'accès personnel d'entreprise

Consultez [GitHub Exemple de serveur d'entreprise](#) pour savoir comment copier un jeton d'accès personnel dans votre presse-papiers. Collez le jeton dans la zone de texte, puis choisissez Save Token (Enregistrer le jeton).

Note

Vous n'avez besoin de saisir et d'enregistrer le jeton d'accès personnel qu'une seule fois. CodeBuild utilise ce jeton dans tous les futurs projets.

Version de la source

Entrez une pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte d'état, entrez la valeur à utiliser pour le `context` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour l'URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

SSL non sécurisé

Sélectionnez Activer le protocole SSL non sécurisé pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub d'entreprise.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

Environnement

Modèle de provisionnement

Effectuez l'une des actions suivantes :

- Pour utiliser des flottes à la demande gérées par AWS CodeBuild, choisissez On-Demand. Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites une fois la construction terminée. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.
- Pour utiliser des flottes de capacité réservée gérées par AWS CodeBuild, choisissez Capacité réservée, puis sélectionnez un nom de flotte. Avec les flottes de capacité réservée, vous configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces

machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

Pour plus d'informations, consultez [Travailler avec une capacité réservée dans AWS CodeBuild](#).

Image de l'environnement

Effectuez l'une des actions suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Managed image (Image gérée), puis effectuez vos sélections pour Système d'exploitation, Exécution, Image et Version d'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager.

Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Calcul

Effectuez l'une des actions suivantes :

- Pour utiliser le calcul EC2, choisissez EC2. Le calcul EC2 offre une flexibilité optimisée lors des exécutions d'actions.
- Pour utiliser le calcul Lambda, choisissez Lambda. Le calcul Lambda offre des vitesses de démarrage optimisées pour vos builds. Lambda permet des builds plus rapides grâce à une

latence de démarrage plus faible. Lambda s'adapte également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Pour plus d'informations, consultez [Utilisation du AWS Lambda calcul dans AWS CodeBuild](#).

Rôle de service

Effectuez l'une des actions suivantes :

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

Note

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

Configuration supplémentaire

Expiration

Spécifiez une valeur, comprise entre 5 minutes et 8 heures, après quoi la CodeBuild génération s'arrête si elle n'est pas terminée. Si les valeurs de heures et minutes sont laissées vides, la valeur par défaut de 60 minutes est utilisée.

privilegié

(Facultatif) Sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos versions obtiennent des privilèges élevés uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre spécification de génération en exécutant les commandes de génération ci-après. N'exécutez pas ces commandes si vous avez choisi une image d'environnement de construction fournie CodeBuild par le support Docker.

Note

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

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise. CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent. CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources des VPC.

Pour plus d'informations, consultez [Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Calcul

Choisissez l'une des options disponibles.

Variables d'environnement

Entrez le nom et la valeur, puis choisissez le type de chaque variable d'environnement à utiliser pour les builds.

Note

CodeBuild définit automatiquement la variable d'environnement pour votre AWS région. Vous devez définir les variables d'environnement suivantes si vous ne les avez pas ajoutées dans votre fichier `buildspec.yml` :


- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`

- IMAGE_TAG

Les utilisateurs de console et d'AWS CLI peuvent voir les variables d'environnement. Si la visibilité de vos variables d'environnement ne vous pose pas de problème, définissez les zones Nom et Valeur, puis définissez Type sur Texte brut.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un identifiant de clé d'AWSaccès, une clé d'accès AWS secrète ou un mot de passe en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager ou.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, choisissez Parameter dans Type. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value, entrez le nom du paramètre tel qu'il est stocké dans le magasin de paramètres Amazon EC2 Systems Manager. Si l'on prend comme exemple un paramètre nommé /CodeBuild/dockerLoginPassword, pour Type, choisissez Parameter (Paramètre). Pour Name (Nom), saisissez LOGIN_PASSWORD. Pour le champ Valeur, saisissez /CodeBuild/dockerLoginPassword.

 Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant /CodeBuild/ par (par exemple /CodeBuild/dockerLoginPassword.). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez Create parameter (Créer un paramètre), puis suivez les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre /CodeBuild/ tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager. Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters` l'action. Si vous avez sélectionné Nouveau rôle

de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de paramètres qui ne commencent pas par `/CodeBuild/`. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres sous l'espace de `/CodeBuild/` noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration `buildspec` a la priorité la plus faible.

Si vous utilisez Secrets Manager, pour Type, choisissez Secrets Manager. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value (Valeur), saisissez un `reference-key` à l'aide du modèle `secret-id:json-key:version-stage:version-id`. Pour plus d'informations, consultez [Secrets Manager reference-key in the buildspec file](#).

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser `secretsmanager:GetSecretValue` action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par `/CodeBuild/`. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de `/CodeBuild/` noms dans le Gestionnaire de secrets.

Spécifications de construction

Spécifications de construction

Effectuez l'une des actions suivantes :

- Si votre code source inclut un fichier `buildspec`, choisissez Utiliser un fichier `buildspec`. Par défaut, CodeBuild recherche un fichier nommé `buildspec.yml` dans le répertoire racine de code source. Si votre fichier `buildspec` utilise un nom ou un emplacement différent, entrez

son chemin depuis la racine source dans le nom Buildspec (par exemple, `buildspec-two.yml` `configuration/buildspec.yml`). Si le fichier `buildspec` se trouve dans un compartiment S3, il doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier `buildspec` à l'aide de son ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

- Si votre code source ne comprend pas de fichier de spécification de génération ou si vous souhaitez exécuter des commandes de génération différentes de celles spécifiées pour la phase `build` dans le fichier `buildspec.yml` au sein du répertoire racine du code source, choisissez Insérer des commandes de génération. Pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase `build`. Pour plusieurs commandes, séparez celles-ci avec `&&` (par exemple, `mvn test && mvn package`). Pour exécuter des commandes dans d'autres phases, ou si vous avez une longue liste de commandes pour la `build` phase, ajoutez un `buildspec.yml` fichier dans le répertoire racine du code source, ajoutez les commandes au fichier, puis choisissez Utiliser le fichier `buildspec.yml` dans le répertoire racine du code source.

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

Configuration par lots

Vous pouvez exécuter un groupe de builds en une seule opération. Pour plus d'informations, consultez [Batch s'intègre AWS CodeBuild](#).

Définition de la configuration par lots

Sélectionnez cette option pour autoriser les compilations par lots dans ce projet.

Rôle du service Batch

Fournit le rôle de service pour les compilations par lots.

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle de service par lots, choisissez Nouveau rôle de service. Dans Rôle de service, entrez le nom du nouveau rôle.
- Si vous avez un rôle de service par lots, choisissez Rôle de service existant. Dans Rôle de service, choisissez le rôle de service.

Les builds par lots introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est obligatoire car CodeBuild il doit être capable d'appeler les `RetryBuild` actions

`StartBuild`/`StopBuild`, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même que celui qu'ils utilisent dans leur build, pour deux raisons :

- L'attribution du rôle `StartBuild` de construction et `RetryBuild` des autorisations permettrait à une seule version de démarrer d'autres versions via le `buildspec`. `StopBuild`
- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds eux-mêmes puissent contourner ces restrictions.

Type (s) de calcul autorisé (s) pour le lot

Sélectionnez les types de calcul autorisés pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Nombre maximal de builds autorisés par lot

Entrez le nombre maximum de builds autorisés dans le lot. Si un lot dépasse cette limite, il échouera.

Délai d'expiration du Batch

Entrez la durée maximale pendant laquelle la génération par lots doit être terminée.

Combinez des artefacts

Sélectionnez Combiner tous les artefacts du lot en un seul emplacement pour que tous les artefacts du lot soient combinés en un seul emplacement.

Mode de rapport par lots

Sélectionnez le mode de rapport d'état de construction souhaité pour les versions par lots.

Note

Ce champ n'est disponible que lorsque la source du projet est Bitbucket ou GitHub Enterprise GitHub, et l'option Signaler les statuts de construction au fournisseur de source lorsque le début et la fin de vos builds sont sélectionnés sous Source.

Constructions agrégées

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient combinés dans un seul rapport d'état.

Constructions individuelles

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient signalés séparément.

Artefacts

Type

Effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer des artefacts de sortie de génération, choisissez Aucun artefact. Vous pouvez le faire si vous exécutez uniquement des tests de compilation ou si vous souhaitez transférer une image Docker vers un référentiel Amazon ECR.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. (Si vous souhaitez produire un fichier ZIP et que vous voulez que celui-ci ait une extension de fichier, veillez à l'inclure après le nom de fichier ZIP.)
 - Sélectionnez Activer la gestion sémantique des versions si vous voulez qu'un nom spécifié dans le fichier buildspec remplace le nom spécifié dans la console. Le nom figurant dans un fichier buildspec est calculé au moment de la génération et utilise le langage de commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, appspec . yml , target/my-app . jar). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).

- Si vous ne souhaitez pas que vos artefacts de génération soient chiffrés, choisissez Remove artifacts encryption (Supprimer le chiffrement des artefacts).

Pour chaque ensemble d'artefacts secondaire que vous souhaitez :

1. Pour Artifact identifier (Identifiant d'artefact), saisissez une valeur de moins de 128 caractères et contenant uniquement des caractères alphanumériques et des traits de soulignement.
2. Choisissez Add artifact (Ajouter un artefact).
3. Suivez les étapes précédentes pour configurer vos artefacts secondaires.
4. Choisissez Save artifact (Enregistrer l'artefact).

Configuration supplémentaire

Clé de chiffrement

(Facultatif) Effectuez l'une des actions suivantes :

- Pour utiliser le Clé gérée par AWS for Amazon S3 dans votre compte afin de chiffrer les artefacts de sortie du build, laissez la clé de chiffrement vide. Il s'agit de l'option par défaut.
- Pour utiliser une clé gérée par le client pour chiffrer les artefacts de sortie de génération, dans Clé de chiffrement, entrez l'ARN de la clé KMS. Utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Type de cache

Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :

- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

⚠ Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

ℹ Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Mise en cache de génération dans AWS CodeBuild](#).

Journaux

Choisissez les journaux que vous souhaitez créer. Vous pouvez créer des CloudWatch journaux Amazon Logs, des journaux Amazon S3 ou les deux.

CloudWatch

Si vous souhaitez obtenir CloudWatch les journaux Amazon Logs :

CloudWatch journaux

Sélectionnez Journaux CloudWatch .

Nom du groupe

Entrez le nom de votre groupe de CloudWatch journaux Amazon Logs.

Nom du stream

Entrez le nom de votre flux de journal Amazon CloudWatch Logs.

S3

Si vous souhaitez obtenir les journaux Amazon S3 :

Journaux S3

Sélectionnez Journaux S3.

Compartiment

Choisissez le nom du compartiment S3 pour vos journaux.

Préfixe de chemin

Entrez le préfixe de vos journaux.

Désactiver le chiffrement des journaux S3

Sélectionnez si vous ne voulez pas que vos journaux S3 soient chiffrés.

Création d'un projet de génération (AWS CLI)

Pour plus d'informations sur l'utilisation du AWS CLI with CodeBuild, consultez le [Référence des commandes en ligne](#).

Pour créer un projet de CodeBuild construction à l'aide de AWS CLI, vous devez créer une structure de [projet](#) au format JSON, renseigner la structure et appeler la [create-project](#) commande pour créer le projet.

Créez le fichier JSON

Créez un fichier JSON squelette à l'aide de la [create-project](#) commande, en utilisant l'`--generate-cli-skeleton` option :

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Cela crée un fichier JSON avec le chemin et le nom de fichier spécifiés par `<json-file>`.

Renseignez le fichier JSON

Modifiez les données JSON comme suit et enregistrez vos résultats.

```
{  
  "name": "<project-name>",  
  "description": "<description>",
```

```

"source": {
  "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
"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" |
"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",
  "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>",
"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-identifiant>",

```

```
    "mountOptions": "<efs-mount-options>"
  }
],
"buildBatchConfig": {
  "serviceRole": "<batch-service-role>",
  "combineArtifacts": <combine-artifacts>,
  "restrictions": {
    "maximumBuildsAllowed": <max-builds>,
    "computeTypesAllowed": [
      "<compute-type>"
    ]
  },
  "timeoutInMins": <batch-timeout>,
  "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
},
"concurrentBuildLimit": <concurrent-build-limit>
}
```

Remplacez les éléments suivants :

nom

Obligatoire. Nom de ce projet de génération. Ce nom doit être unique pour tous les projets de génération de votre compte AWS.

description

Facultatif. Description de ce projet de génération.

source

Obligatoire. Un [ProjectSource](#) objet qui contient des informations sur les paramètres du code source de ce projet de build. Après avoir ajouté un objet source, vous pouvez ajouter jusqu'à 12 autres sources à l'aide de `addSource`. Il s'agit notamment des paramètres suivants :

source/type

Obligatoire. Type de référentiel qui contient le code source à générer. Les valeurs valides sont les suivantes :

- CODECOMMIT
- CODEPIPELINE
- GITHUB

- GITHUB_ENTERPRISE
- BITBUCKET
- S3
- NO_SOURCE

Si vous utilisez NO_SOURCE, buildspec ne peut pas être un fichier, car le projet n'a pas de source. En revanche, vous devez utiliser l'attribut `buildspec` afin de spécifier une chaîne au format YAML pour votre buildspec. Pour plus d'informations, consultez [Exemple de projet sans source](#).

source/ lieu

Obligatoire sauf si vous avez défini `<source-type>` sur CODEPIPELINE. Emplacement du code source pour le type de référentiel spécifié.

- Pour CodeCommit, l'URL du clone HTTPS vers le référentiel qui contient le code source et le fichier buildspec (par exemple,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Pour Amazon S3, le nom du compartiment d'entrée de construction, suivi du chemin et du nom du fichier ZIP contenant le code source et les spécifications de construction. Par exemple :
 - Pour un fichier ZIP situé à la racine du compartiment d'entrée : `<bucket-name>/<object-name>.zip`.
 - Pour un fichier ZIP situé dans un sous-dossier du compartiment d'entrée : `<bucket-name>/<subfolder-path>/<object-name>.zip`.
- Pour GitHub, l'URL du clone HTTPS vers le référentiel contenant le code source et le fichier buildspec. L'URL doit contenir github.com. Vous devez connecter votre AWS compte à votre GitHub compte. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.
 1. Sur la page GitHub Autoriser l'application, dans la section Accès à l'organisation, choisissez Demander l'accès à côté de chaque référentiel auquel vous souhaitez CodeBuild accéder dans le.
 2. Choisissez Authorize application. (Une fois connecté à votre GitHub compte, vous n'avez pas besoin de terminer la création du projet de construction. Vous pouvez fermer la CodeBuild console.)
- Pour GitHub Enterprise Server, URL du clone HTTP ou HTTPS du référentiel contenant le code source et le fichier buildspec. Vous devez également connecter votre AWS compte à votre compte GitHub Enterprise Server. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.
 1. Créez un jeton d'accès personnel dans GitHub Enterprise Server.

2. Copiez ce jeton dans votre presse-papiers afin de pouvoir l'utiliser lors de la création de votre CodeBuild projet. Pour plus d'informations, consultez la section [Création d'un jeton d'accès personnel pour la ligne de commande](#) sur le site Web d' GitHub aide.
 3. Lorsque vous utilisez la console pour créer votre CodeBuild projet, dans Source, pour Source provider, choisissez GitHubEnterprise.
 4. Pour Personal Access Token, collez le jeton copié dans le Presse-papiers. Choisissez Save Token. Votre CodeBuild compte est désormais connecté à votre compte GitHub Enterprise Server.
- Pour Bitbucket, l'URL de clone HTTPS vers le référentiel qui contient le code source et le fichier buildspec. L'URL doit contenir bitbucket.org. Vous devez également connecter votre compte AWS à votre compte Bitbucket. Pour ce faire, utilisez la CodeBuild console pour créer un projet de génération.
 1. Lorsque vous utilisez la console pour vous connecter (ou vous reconnecter) à Bitbucket, dans la page Bitbucket Confirm access to your account, choisissez Grant access. (Une fois connecté à votre compte Bitbucket, vous n'avez pas besoin de terminer la création du projet de build. Vous pouvez fermer la CodeBuild console.)
 - Pour AWS CodePipeline, ne spécifiez pas de location valeur pour source. CodePipeline ignore cette valeur car lorsque vous créez un pipeline dans CodePipeline, vous spécifiez l'emplacement du code source dans le stage Source du pipeline.

source/ gitCloneDepth

Facultatif. Profondeur de l'historique à télécharger. La valeur minimum est de 0. Si cette valeur est égale à 0, supérieure à 25 ou non spécifiée, l'historique complet est téléchargé avec chaque projet de génération. Si votre type de source est Amazon S3, cette valeur n'est pas prise en charge.

source/ buildspec

Facultatif. Définition de spécification de génération ou fichier à utiliser. Si cette valeur n'est pas fournie ou qu'elle est définie sur une chaîne vide, le code source doit contenir un fichier `buildspec.yml` dans son répertoire racine. Si cette valeur est définie, il peut s'agir d'une définition de buildspec intégrée, du chemin d'accès à un autre fichier buildspec relatif au répertoire racine de votre source principale ou du chemin d'accès à un compartiment S3. Le compartiment doit être dans la même région AWS que le projet de génération. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Pour plus d'informations, consultez [Nom de fichier buildspec et emplacement de stockage](#).

source/ authentication

Ne pas utiliser. Cet objet est utilisé uniquement par la CodeBuild console.

source/ reportBuildStatus

Indique si vous souhaitez envoyer le statut de début et de fin d'une génération à votre fournisseur de source. Si vous le définissez avec un fournisseur de source autre qu' GitHub GitHub Enterprise Server ou Bitbucket, un `invalidInputException` est émis.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

source/ buildStatusConfig

Contient des informations qui définissent la manière dont le CodeBuild projet de construction communique l'état de la construction au fournisseur source. Cette option n'est utilisée que lorsque le type de source est `GITHUBGITHUB_ENTERPRISE`, ou `BITBUCKET`.

source/buildStatusConfig/contexte

Pour les sources Bitbucket, ce paramètre est utilisé pour le `name` paramètre figurant dans le statut de validation de Bitbucket. Pour les GitHub sources, ce paramètre est utilisé pour le `context` paramètre figurant dans l'état de GitHub validation.

Par exemple, vous pouvez faire en sorte qu'ils `context` contiennent le numéro de version et le déclencheur webhook à l'aide des variables d' CodeBuildenvironnement :

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -  
${CODEBUILD_WEBHOOK_TRIGGER}
```

Il en résulte que le contexte apparaît comme suit pour la version #24 déclenchée par un événement de pull request du webhook :

```
AWS CodeBuild sample-project Build #24 - pr/8
```

source/buildStatusConfig/URL cible

Pour les sources Bitbucket, ce paramètre est utilisé pour le `url` paramètre figurant dans le statut de validation de Bitbucket. Pour les GitHub sources, ce paramètre est utilisé pour le `target_url` paramètre figurant dans l'état de GitHub validation.

Par exemple, vous pouvez `targetUrl` définir le statut sur `https://aws.amazon.com/codebuild/<path to build>` et le statut de validation sera lié à cette URL.

Vous pouvez également inclure des variables d' CodeBuild environnement dans le `targetUrl` pour ajouter des informations supplémentaires à l'URL. Par exemple, pour ajouter la région de construction à l'URL, définissez la valeur `targetUrl` sur :

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Si la région de construction l'est `us-east-2`, elle s'étendra à :

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

source/ gitSubmodulesConfig

Facultatif. Informations sur la configuration des sous-modules Git. Utilisé uniquement avec CodeCommit GitHub, GitHub Enterprise Server et Bitbucket.

source/gitSubmodulesConfig/ FetchSubmodules

Définissez `fetchSubmodules` sur `true` si vous souhaitez inclure les sous-modules Git dans votre référentiel. Les sous-modules Git inclus doivent être configurés en tant que HTTPS.

source/ InsecureSsl

Facultatif. Utilisé uniquement avec GitHub Enterprise Server. Définissez cette valeur sur `true` pour ignorer les avertissements TLS lors de la connexion au référentiel de votre projet GitHub Enterprise Server. La valeur par défaut est `false`. `InsecureSsl` doit être utilisé à des fins de test uniquement. Cette option ne doit pas être utilisée dans un environnement de production.

source/ Identifiant de source

Identifiant défini par l'utilisateur pour la source du projet. Facultatif pour la source principale. Nécessaire pour les sources secondaires.

Sources secondaires

Facultatif. Tableau d'[ProjectSource](#) objets contenant des informations sur les sources secondaires d'un projet de construction. Vous pouvez ajouter jusqu'à 12 sources secondaires. Les `secondarySources` objets utilisent les mêmes propriétés que celles utilisées par l'objet. Dans un objet source secondaire, le `sourceIdentifier` est obligatoire.

secondarySourceVersions

Facultatif. Tableau d'objets [ProjectSourceVersion](#). Si `secondarySourceVersions` est spécifié au niveau de la build, ils ont priorité sur cela.

Version de la source

Facultatif. Version de l'entrée de compilation à créer pour ce projet. Si elle n'est pas spécifiée, la dernière version est utilisée. Si cette valeur est spécifiée, elle doit être l'une des suivantes :

- Pour CodeCommit, l'ID de validation, la branche ou le tag Git à utiliser.
- Pour GitHub, l'ID de validation, l'ID de pull request, le nom de branche ou le nom de balise correspondant à la version du code source que vous souhaitez créer. Si un ID de demande d'extraction est spécifié, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Pour Bitbucket, l'ID de validation, le nom de branche ou le nom de balise qui correspond à la version du code source à générer. Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. S'il n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Pour Amazon S3, ID de version de l'objet qui représente le fichier ZIP d'entrée de génération à utiliser.

Si `sourceVersion` est spécifié à la build, cette version est prioritaire sur cette `sourceVersion` (au niveau du projet). Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

artefacts

Obligatoire. Un [ProjectArtifacts](#) objet qui contient des informations sur les paramètres d'artefact de sortie de ce projet de construction. Après avoir ajouté un objet `artifacts`, vous pouvez ajouter jusqu'à 12 autres artefacts à l'aide de `artifacts`. Il s'agit notamment des paramètres suivants :

artefacts/type

Obligatoire. Le type d'objet de sortie de la build. Les valeurs valides sont :

- `CODEPIPELINE`
- `NO_ARTIFACTS`

- S3

artefacts/ emplacement

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Le nom du bucket de sortie que vous avez créé ou identifié dans les prérequis.

artefacts/ chemin

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Le chemin d'accès au compartiment de sortie où placer le fichier ou le dossier ZIP. Si vous ne spécifiez aucune valeur pour `path`, CodeBuild utilise `namespaceType` (si spécifié) et `name` pour déterminer le chemin et le nom du fichier ZIP ou du dossier de sortie de compilation. Par exemple, si vous spécifiez `MyPath` pour `path` et `MyArtifact.zip` pour `name`, le chemin et le nom seront `MyPath/MyArtifact.zip`.

artefacts/ NameSpaceType

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

L'espace de noms du fichier ZIP ou du dossier de sortie de compilation. Les valeurs valides sont `BUILD_ID` et `NONE`. Utilisez `BUILD_ID` pour insérer l'ID de génération dans le chemin du fichier ZIP ou dossier de sortie de génération. Dans le cas contraire, utilisez `NONE`. Si vous ne spécifiez aucune valeur pour `namespaceType`, CodeBuild utilise `path` (si spécifié) et `name` pour déterminer le chemin et le nom du fichier ZIP ou du dossier de sortie de compilation. Par exemple, si vous spécifiez `MyPath BUILD_ID` pour `path namespaceType`, pour et `MyArtifact.zip` pour `name`, le chemin et le nom seront `MyPath/build-ID/MyArtifact.zip`.

artefacts/nom

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Le nom du fichier ZIP ou du dossier de sortie de compilation qu'il contient `location`. Par exemple, si vous spécifiez `MyPath` pour `path` et `MyArtifact.zip` pour `name`, le chemin et le nom seront `MyPath/MyArtifact.zip`.

artefacts/ overrideArtifactName

Utilisé uniquement avec le type d'artefact S3. Non utilisé pour les autres types d'artefacts.

Facultatif. S'il est défini sur `true`, le nom spécifié dans le `artifacts` bloc du fichier `buildspec` est remplacé. Pour plus d'informations, consultez [Référence de spécification de construction pour CodeBuild](#).

artefacts/emballages

Utilisé uniquement avec le type d'S3artefact. Non utilisé pour les autres types d'artefacts.

Facultatif. Spécifie comment emballer les artefacts. Les valeurs autorisées sont les suivantes :

NONE

Créez un dossier contenant les artefacts de construction. C'est la valeur par défaut.

ZIP

Créez un fichier ZIP contenant les artefacts de construction.

secondaryArtifacts

Facultatif. Tableau d'[ProjectArtifacts](#)objets contenant des informations sur les paramètres des artefacts secondaires d'un projet de construction. Vous pouvez ajouter jusqu'à 12 artefacts secondaires. `secondaryArtifacts` utilise un grand nombre de paramètres communs à l'objet .

cache

Obligatoire. Un [ProjectCache](#)objet qui contient des informations sur les paramètres de cache de ce projet de génération. Pour plus d'informations, consultez [Mise en cache de génération](#).

environment

Obligatoire. Un [ProjectEnvironment](#)objet qui contient des informations sur les paramètres de l'environnement de construction de ce projet. Ces paramètres sont les suivants :

environnement/type

Obligatoire. Le type d'environnement de génération. Pour plus d'informations, reportez-vous à la section [Type](#) dans le Guide de référence de CodeBuild l'API.

environnement/image

Obligatoire. Identifiant d'image Docker utilisé par cet environnement de génération. En général, cet identificateur est exprimé sous la forme *image-name:balise*. Par exemple, dans le référentiel Docker CodeBuild utilisé pour gérer ses images Docker, cela pourrait être le cas. `aws/codebuild/standard:5.0` Dans Docker Hub, `maven:3.3.9-jdk-8`. Dans Amazon ECR, `account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag`. Pour plus d'informations, consultez [Images Docker fournies par CodeBuild](#).

environnement/ ComputeType

Obligatoire. Spécifie les ressources de calcul utilisées par cet environnement de génération. Pour plus d'informations, consultez [ComputeType](#) dans CodeBuild le Guide de référence des API.

environnement/certificat

Facultatif. L'ARN du compartiment Amazon S3, le préfixe de chemin et la clé d'objet contenant le certificat codé PEM. La clé d'objet peut être juste le fichier .pem ou un fichier .zip contenant le certificat codé PEM. Par exemple, si le nom de votre compartiment Amazon S3 est `<my-bucket>`, votre préfixe de chemin est `<cert>` et le nom de votre clé d'objet est `<certificate.pem>`, les formats acceptables pour certificate sont `<my-bucket/cert/certificate.pem>` ou `arn:aws:s3:::<my-bucket/cert/certificate.pem>`.

environnement/ Variables d'environnement

Facultatif. Tableau d'[EnvironmentVariable](#) objets contenant les variables d'environnement que vous souhaitez spécifier pour cet environnement de génération. Chaque variable d'environnement est exprimée sous la forme d'un name objet contenant a name, value, et type. value type

La console et AWS CLI les utilisateurs peuvent voir toutes les variables d'environnement. Si vous n'avez aucune inquiétude quant à la visibilité de votre variable d'environnement, définissez name et value et réglez type sur PLAINTEXT.

Nous vous recommandons de stocker les variables d'environnement contenant des valeurs sensibles, telles qu'un identifiant de clé d'AWS accès, une clé d'accès AWS secrète ou un mot de passe, en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager ou. Pour name, pour ce paramètre stocké, définissez un identifiant CodeBuild à référencer.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, value pour, définissez le nom du paramètre tel qu'il est stocké dans le Parameter Store. Définissez type sur PARAMETER_STORE. À l'aide d'un paramètre nommé à /CodeBuild/dockerLoginPassword titre d'exemple, définissez name sur LOGIN_PASSWORD. Définissez value sur /CodeBuild/dockerLoginPassword. Définissez type sur PARAMETER_STORE.

Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant / CodeBuild/ par (par exemple /CodeBuild/dockerLoginPassword,). Vous pouvez

utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez Create parameter (Créer un paramètre), puis suivez les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre `/CodeBuild/` tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters` l'action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de paramètres qui ne commencent pas par `/CodeBuild/`. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres sous l'espace de `/CodeBuild/` noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration `buildspec` a la priorité la plus faible.

Si vous utilisez Secrets Manager, pour `value`, définissez le nom du paramètre tel qu'il est enregistré dans Secrets Manager. Définissez `type` sur `SECRETS_MANAGER`. À l'aide d'un secret nommé `/CodeBuild/dockerLoginPassword` titre d'exemple, définissez `name` sur `LOGIN_PASSWORD`. Définissez `value` sur `/CodeBuild/dockerLoginPassword`. Définissez `type` sur `SECRETS_MANAGER`.

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser `secretsmanager:GetSecretValue`action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par `/CodeBuild/`. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de /CodeBuild/ noms dans le Gestionnaire de secrets.

Environnement/RegistryCredential

Facultatif. [RegistryCredential](#) Objet qui spécifie les informations d'identification permettant d'accéder à un registre Docker privé.

Environnement/registration/accréditation

Spécifie l'ARN ou le nom des informations d'identification créées à l'aide de AWS Managed Services. Vous pouvez utiliser le nom des informations d'identification uniquement si elles existent dans votre région actuelle.

Environnement/accréditation de registre/fournisseur d'informations d'identification

La seule valeur valide est SECRETS_MANAGER.

Lorsque cela est défini :

- `imagePullCredentials` doit être défini sur `SERVICE_ROLE`.
- L'image ne peut pas être une image sélectionnée ou une image Amazon ECR.

environnement/ Type imagePullCredentials

Facultatif. Le type d'informations d'identification CodeBuild utilisé pour extraire des images dans votre build. Deux valeurs sont valides :

CONSTRUCTION DE CODE

`CODEBUILD` indique qu'il CodeBuild utilise ses propres informations d'identification. Vous devez modifier votre politique de dépôt Amazon ECR pour faire confiance au principal du CodeBuild service.

RÔLE_SERVICE

Spécifie qui CodeBuild utilise le rôle de service de votre projet de construction.

Lorsque vous utilisez une image de registre privé ou entre comptes, vous devez utiliser les informations d'identification de `SERVICE_ROLE`. Lorsque vous utilisez une image CodeBuild sélectionnée, vous devez utiliser des `CODEBUILD` informations d'identification.

environnement/ PrivilegedMode

Définissez cette valeur `true` uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre fichier `buildspec` en exécutant les commandes de génération ci-après. N'exécutez pas ces commandes si vous avez spécifié une image d'environnement de construction fournie CodeBuild par le support Docker.

Note

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

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

serviceRole

Obligatoire. L'ARN du rôle de service est CodeBuild utilisé pour interagir avec les services pour le compte de l'utilisateur (par exemple, `arn:aws:iam::account-id:role/role-name`).

timeoutInMinutes

Facultatif. Nombre de minutes, compris entre 5 et 480 (8 heures), au-delà desquelles CodeBuild arrête la génération si celle-ci n'est pas terminée. À défaut de précision, la valeur par défaut 60 est utilisée. Pour déterminer si et quand une compilation CodeBuild a été arrêtée en raison d'un délai d'attente, exécutez la `batch-get-builds` commande. Pour déterminer si la génération s'est arrêtée, recherchez dans la sortie une valeur `buildStatus` pour `FAILED`. Pour déterminer quand la génération a expiré, recherchez dans la sortie la valeur `endTime` associée à une valeur `phaseStatus` pour `TIMED_OUT`.

queuedTimeoutInMinutes

Facultatif. Le nombre de minutes, compris entre 5 et 480 (8 heures), après quoi la construction CodeBuild s'arrête si elle est toujours en file d'attente. À défaut de précision, la valeur par défaut 60 est utilisée.

encryptionKey

Facultatif. Alias ou ARN du fichier AWS KMS key utilisé par CodeBuild pour chiffrer la sortie de compilation. Si vous spécifiez un alias, utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID` ou, s'il existe un alias, utilisez le format `alias/key-alias`. Si elle n'est pas spécifiée, la clé KMS AWS gérée pour Amazon S3 est utilisée.

balises

Facultatif. Un tableau d'objets [Tag](#) qui fournissent les balises que vous souhaitez associer à ce projet de construction. Vous pouvez spécifier jusqu'à 50 balises. Ces balises peuvent être utilisées par n'importe quel AWS service qui prend en charge les balises de projet de CodeBuild construction. Chaque balise est exprimée sous la forme d'un objet avec a key et value a.

vpcConfig

Facultatif. [VpcConfig](#)Objet contenant des informations sur la configuration VPC de votre projet. Pour plus d'informations, consultez [UtiliserAWS CodeBuildavec Amazon Virtual Private Cloud](#).

Ces propriétés incluent :

vpclId

Obligatoire. L'ID VPC qui CodeBuild utilise. Pour obtenir la liste de tous les identifiants de VPC dans votre région, exécutez la commande suivante :

```
aws ec2 describe-vpcs --region <region-ID>
```

sous-réseaux

Obligatoire. Un tableau d'identifiants de sous-réseau qui inclut les ressources utilisées par CodeBuild. Pour obtenir ces identifiants, exécutez la commande suivante :

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

securityGroupIds

Obligatoire. Tableau d'identifiants de groupes de sécurité utilisés CodeBuild pour autoriser l'accès aux ressources du VPC. Pour obtenir ces identifiants, exécutez la commande suivante :

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

badgeEnabled

Facultatif. Spécifie s'il faut inclure des badges de construction dans votre CodeBuild projet. Définissez sur `true` pour activer la création de badges, ou `false` autre. Pour plus d'informations, consultez [Créez un échantillon de badges avec CodeBuild](#).

Configuration des journaux

Un [LogsConfig](#) objet qui contient des informations sur l'emplacement des journaux de cette version.

Configuration des journaux/ cloudWatchLogs

[CloudWatchLogsConfig](#) Objet contenant des informations sur le transfert de CloudWatch journaux vers Logs.

LogsConfig/ S3 Logs

Un LogsConfig objet [S3](#) qui contient des informations sur le transfert de journaux vers Amazon S3.

fileSystemLocations

Facultatif. Tableau d'[ProjectFileSystemsLocation](#) objets contenant des informations sur votre configuration Amazon EFS.

buildBatchConfig

Facultatif. L'`buildBatchConfig` objet est une [ProjectBuildBatchConfig](#) structure qui contient les informations de configuration de génération par lots pour le projet.

buildBatchConfig/ServiceRole

L'ARN du rôle de service pour le projet de génération par lots.

buildBatchConfig/Combinez des artefacts

Valeur booléenne qui indique s'il faut combiner les artefacts de construction pour la construction par lots dans un seul emplacement d'artefact.

buildBatchConfig/restrictions/ maximumBuildsAllowed

Le nombre maximum de builds autorisés.

buildBatchConfig/restrictions/ computeTypesAllowed

Tableau de chaînes qui spécifient les types de calcul autorisés pour la construction par lots.

Reportez-vous [à la section Types de calcul de l'environnement](#) de génération pour ces valeurs.

buildBatchConfig/timeoutInMinutes

Durée maximale, en minutes, pendant laquelle la génération par lots doit être terminée.

buildBatchConfig/batchReportMode

Spécifie comment les rapports d'état de compilation sont envoyés au fournisseur source pour la compilation par lots. Les valeurs valides sont les suivantes :

REPORT_AGGREGATED_BATCH

(Par défaut) Regrouper tous les états de compilation dans un seul rapport d'état.

REPORT_INDIVIDUAL_BUILDS

Envoyer un rapport d'état distinct pour chaque compilation individuelle.

concurrentBuildLimit

Nombre maximal de générations simultanées autorisées pour ce projet.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Création du projet

Pour créer le projet, réexécutez la [create-project](#) commande en passant votre fichier JSON :

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

En cas de succès, la représentation JSON d'un objet [Project](#) apparaît dans la sortie de la console. Voir la [syntaxe de CreateProject réponse](#) pour un exemple de ces données.

À l'exception du nom de projet de génération, vous pouvez modifier les paramètres du projet de génération ultérieurement. Pour plus d'informations, consultez [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).

Pour démarrer l'exécution d'une génération, consultez [Exécution d'une génération \(AWS CLI\)](#).

Si votre code source est stocké dans un GitHub référentiel et que vous CodeBuild souhaitez le reconstruire chaque fois qu'une modification de code est envoyée au référentiel, consultez [Lancement automatique de l'exécution de générations \(AWS CLI\)](#).

Création d'un projet de génération (kits SDK AWS)

Pour de plus amples informations sur l'utilisation d'AWS CodeBuild avec le kit SDK AWS, consultez la [Référence des kits SDK et des outils AWS](#).

Création d'un projet de génération (AWS CloudFormation)

Pour obtenir des informations sur l'utilisation d'AWS CodeBuild avec AWS CloudFormation, voir [le AWS CloudFormation modèle pour CodeBuild](#) dans le AWS CloudFormation Guide de l'utilisateur.

Création d'une règle de notification

Vous pouvez utiliser des règles de notification pour informer les utilisateurs de changements importants, notamment des succès et des échecs de création. Les règles de notification spécifient à la fois les événements et le sujet Amazon SNS utilisé pour envoyer des notifications. Pour plus d'informations, consultez [Que sont les notifications ?](#)

Vous pouvez utiliser la console ou l'AWS CLI pour créer des règles de notification pour AWS CodeBuild.

Pour créer une règle de notification (console)

1. Connectez-vous à la CodeBuild console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Choisissez Build (Générer), puis Build projects (Générer des projets), puis sélectionnez un projet de création auquel vous souhaitez ajouter des notifications.

3. Sur la page du projet de création, choisissez Notify (Notifier), puis Create notification rule (Créer une règle de notification). Vous pouvez également accéder à la page Settings (Paramètres) du projet de génération et choisir Create notification rule (Créer une règle de notification).
4. Dans Notification name (Nom de la notification), saisissez le nom de la règle.
5. Dans Type de détail, choisissez Basic si vous souhaitez que seules les informations fournies à Amazon soient EventBridge incluses dans la notification. Choisissez Complet si vous souhaitez inclure les informations fournies à Amazon EventBridge et les informations susceptibles d'être fournies par le CodeBuild ou le gestionnaire de notifications.

Pour plus d'informations, consultez [Présentation du contenu des notifications et de la sécurité](#).

6. Dans Événements qui déclenchent des notifications, sélectionnez les événements pour lesquels vous souhaitez envoyer des notifications. Pour de plus amples informations, veuillez consulter [Événements pour les règles de notification sur les projets de génération](#).
7. Dans Targets (Cibles), effectuez l'une des actions suivantes :
 - Si vous avez déjà configuré une ressource à utiliser avec les notifications, dans Choose target type (Choisir le type de cible), choisissez AWS Chatbot (Slack) ou SNS topic (Rubrique SNS). Dans Choisir une cible, choisissez le nom du client (pour un client Slack configuré dans AWS Chatbot) ou le nom de ressource Amazon (ARN) de la rubrique Amazon SNS (pour les rubriques Amazon SNS déjà configurées avec la politique requise pour les notifications).
 - Si vous n'avez pas configuré de ressource à utiliser avec les notifications, choisissez Create target (Créer une cible), puis SNS topic (Rubrique SNS). Donnez un nom à la rubrique après codestar-notifications-, puis choisissez Create (Créer).

Note

- Si vous créez la rubrique Amazon SNS dans le cadre de la création de la règle de notification, la stratégie qui permet à la fonctionnalité de notifications de publier des événements dans la rubrique est appliquée automatiquement. L'utilisation d'une rubrique créée pour les règles de notification vous permet de vous assurer que vous n'abonnez que les utilisateurs qui doivent recevoir des notifications sur ce référentiel.
- Vous ne pouvez pas créer un client AWS Chatbot dans le cadre de la création d'une règle de notification. Si vous choisissez AWS Chatbot (Slack), vous verrez un bouton vous demandant de configurer un client dans AWS Chatbot. Le choix de cette option

provoque l'ouverture de la console AWS Chatbot. Pour de plus amples informations, veuillez consulter [Configurer les intégrations entre les notifications et AWS Chatbot](#).

- Si vous souhaitez utiliser une rubrique Amazon SNS existante comme cible, vous devez ajouter la politique requise pour les AWS CodeStar notifications en plus de toute autre politique susceptible d'exister pour cette rubrique. Pour de plus amples informations, veuillez consulter [Configurer les rubriques Amazon SNS existantes pour les notifications](#) et [Présentation du contenu des notifications et de la sécurité](#).

8. Pour terminer la création de la règle, choisissez Submit (Soumettre).
9. Vous devez inscrire les utilisateurs à la rubrique Amazon SNS relative à la règle pour qu'ils puissent recevoir des notifications. Pour plus d'informations, consultez [Abonner des utilisateurs aux rubriques Amazon SNS qui sont des cibles](#). Vous pouvez également configurer l'intégration entre les notifications et envoyer des notifications AWS Chatbot aux forums de discussion Amazon Chime. Pour plus d'informations, consultez [Configurer l'intégration entre les notifications et AWS Chatbot](#).

Pour créer une règle de notification (AWS CLI)

1. Dans un terminal ou une invite de commandes, exécutez la commande `create-notification rule` pour générer le squelette JSON :

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

Vous pouvez donner au fichier le nom de votre choix. Dans cet exemple, le fichier est nommé *rule.json*.

2. Ouvrez le fichier JSON dans un éditeur de texte brut et modifiez-le pour y inclure la ressource, les types d'événements et la cible que vous souhaitez pour la règle. L'exemple suivant montre une règle de notification nommée **MyNotificationRule** pour un projet de construction nommé *MyBuildProject* dans un AWS compte avec l'ID *123456789012*. Les notifications sont envoyées avec le type de détail complet à une rubrique Amazon SNS nommée *codestar-notifications*, lorsque les builds sont réussis : MyNotificationTopic

```
{  
  "Name": "MyNotificationRule",  
  "EventIds": [  
    "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"
}
```

Sauvegardez le fichier.

3. À l'aide du fichier que vous venez de modifier, à partir du terminal ou de la ligne de commande, exécutez à nouveau la commande `create-notification-rule` pour créer la règle de notification :

```
aws codestarnotifications create-notification-rule --cli-input-json
file://rule.json
```

4. En cas de réussite, la commande renvoie l'ARN de la règle de notification, comme suit :

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Affichage d'une liste de noms de projet de génération dans AWS CodeBuild

Vous pouvez utiliser le plugin `AWS CodeBuild console` AWS CLI, ou `AWSSDK` pour afficher une liste des projets de génération dans CodeBuild.

Rubriques

- [Affichage d'une liste de noms de projet de génération \(console\)](#)
- [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#)
- [Affichage d'une liste de noms de projet de génération \(kits SDK AWS\)](#)

Affichage d'une liste de noms de projet de génération (console)

Vous pouvez afficher la liste des projets de génération dans une région AWS dans la console. Les informations comprennent le nom, le fournisseur source, le référentiel, le statut de la dernière génération et la description, le cas échéant.

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projets per page (Projets par page) ou utilisez les flèches avant et arrière.

Affichage d'une liste de noms de projet de génération (AWS CLI)

Exécutez la commande `list-projects` :

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *sort-by* : chaîne facultative utilisée pour indiquer les critères à utiliser pour établir la liste des noms de projet de génération. Les valeurs valides sont les suivantes :
 - `CREATED_TIME` : afficher les noms de projet de génération en fonction de la date à laquelle chaque projet de génération a été créé.
 - `LAST_MODIFIED_TIME` : afficher les noms de projet de génération en fonction du moment où les informations sur chaque projet de génération ont été modifiées pour la dernière fois.
 - `NAME` : afficher les noms de projet de génération en fonction du nom de chaque projet de génération.
- *sort-order* : chaîne facultative utilisée pour indiquer l'ordre dans lequel dresser la liste des projets de génération, en fonction de la valeur de *sort-by*. Les valeurs valides sont `ASCENDING` et `DESCENDING`.

- **jeton suivant** : chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}
```

Affichage d'une liste de noms de projet de génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Affichage des détails d'un projet de génération dans AWS CodeBuild

Vous pouvez utiliser le plugin `aws codebuild console` AWS CLI, ou `aws` kits SDK pour afficher les détails d'un projet de génération dans CodeBuild.

Rubriques

- [Affichage des détails d'un projet de génération \(console\)](#)
- [Affichage des détails d'un projet de génération \(AWS CLI\)](#)
- [Affichage des détails d'un projet de génération \(kits SDK AWS\)](#)

Affichage des détails d'un projet de génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour `Projects per page` (Projets par page) ou utilisez les flèches avant et arrière.

3. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien du projet de génération.
4. Sur la page Projet de génération : ***nom-projet***, choisissez Détails de génération.

Affichage des détails d'un projet de génération (AWS CLI)

Exécutez la commande `batch-get-projects` :

```
aws codebuild batch-get-projects --names names
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- ***noms*** : chaîne obligatoire utilisée pour indiquer un ou plusieurs noms de projets de génération pour lequel afficher les détails. Pour spécifier plusieurs projets de génération, séparez leurs noms par un espace. Vous pouvez spécifier jusqu'à 100 noms de projet de génération. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2
my-other-demo-project
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie. Des ellipses (. . .) sont utilisées pour représenter les données omises par souci de concision.

```
{
  "projectsNotFound": [
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": codebuild-demo-project,
      ...
    },
    {
      ...
      "name": codebuild-demo-project2",
      ...
    }
  ]
}
```

Dans la sortie précédente, le tableau `projectsNotFound` répertorie les noms de projet de génération qui ont été spécifiés, mais pour lesquels aucune information n'a été trouvée. Le tableau `projects` répertorie les détails pour chaque projet de génération pour lequel des informations ont été trouvées. Des détails de projet de génération ont été omis dans la sortie précédente par souci de concision. Pour plus d'informations, consultez la sortie de [Création d'un projet de génération \(AWS CLI\)](#).

Lebatch-get-projects ne prend pas en charge le filtrage de certaines valeurs de propriétés, mais vous pouvez écrire un script qui énumère les propriétés d'un projet. Par exemple, le script shell Linux suivant énumère les projets de la région actuelle pour le compte courant et imprime l'image utilisée par chaque projet.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.

imageName=""

function getImageName(){
    local environmentValues=(${1//$\t/ })
    imageName=${environmentValues[1]}
}

function processProjectInfo() {
    local projectInfo=$1

    while IFS=$'\t' read -r section value; do
        if [[ "$section" == *"ENVIRONMENT"* ]]; then
            getImageName "$value"
        fi
    done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
    if [[ "$projectName" != *"PROJECTS"* ]]; then
        echo "====="

        # Get the detailed information for the project.
        projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

        processProjectInfo "$projectInfo"

        printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
    fi
done
```

```
done
```

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Affichage des détails d'un projet de génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Mise en cache de génération dans AWS CodeBuild

Vous pouvez faire des économies lors de vos générations de projet en utilisant un cache. Un cache peut stocker les éléments réutilisables de votre environnement de génération et les utiliser sur plusieurs générations. Votre projet de génération peut utiliser l'un des deux types de mise en cache suivants : Amazon S3 ou local. Si vous utilisez un cache local, vous devez choisir un ou plusieurs des trois modes de cache : le cache source, le cache de couche Docker et le cache personnalisé.

Note

Le mode de cache de couche Docker est disponible pour l'environnement Linux uniquement. Si vous choisissez ce mode, vous devez exécuter votre build en mode privilégié. CodeBuild le mode privilégié des projets accorde à son conteneur l'accès à tous les appareils. Pour de plus amples informations, veuillez consulter [Privilège d'exécution et fonctionnalités Linux](#) sur le site web Docker Docs.

Rubriques

- [Mise en cache d'Amazon S3](#)
- [Mise en cache locale](#)

Mise en cache d'Amazon S3

La mise en cache Amazon S3 stocke le cache dans un compartiment Amazon S3 disponible sur plusieurs hôtes de build. C'est une bonne option pour les artefacts de construction de petite ou moyenne taille qui sont plus coûteux à créer qu'à télécharger. Ce n'est pas la meilleure option pour des artefacts de génération de grande taille, car le temps de transfert sur votre réseau peut être

important et avoir un impact sur les performances de génération. Ce n'est pas non plus la meilleure option si vous utilisez des couches Docker.

Mise en cache locale

La mise en cache locale stocke un cache localement sur un hôte de génération. Le cache est disponible uniquement sur cet hôte de génération. C'est une bonne option pour les artefacts de build de taille intermédiaire à grande, car le cache est immédiatement disponible sur l'hôte de build. Ce n'est pas la meilleure option si vos builds sont rares. Cela signifie que les performances de génération ne sont pas affectés par le temps de transfert réseau.

Si vous choisissez la mise en cache locale, vous devez choisir un ou plusieurs des modes de cache suivants :

- Le mode de cache source met en cache des métadonnées Git pour des sources principales et secondaires. Lorsque le cache est créé, les générations suivantes extraient uniquement les modifications apportées entre les validations. Ce mode est un bon choix pour les projets avec un répertoire de travail propre et une source correspondant à un référentiel Git de grande taille. Si vous choisissez cette option et que votre projet n'utilise pas de référentiel Git (AWS CodeCommit, GitHub, GitHub Enterprise Server ou Bitbucket), l'option est ignorée.
- Le mode de cache de couche Docker met en cache des couches Docker existantes. Ce mode est un bon choix pour les projets qui génèrent ou extraient des images Docker de taille importante. Il permet d'éviter les problèmes de performances causées par l'extraction d'images Docker de grande taille à partir du réseau.

Note

- Vous pouvez utiliser un cache de couche Docker uniquement dans l'environnement Linux.
- L'indicateur `privileged` doit être défini de sorte que votre projet dispose des autorisations Docker requises.

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

- Vous devez prendre en compte les implications de sécurité avant d'utiliser un cache de couche Docker.

- Le mode de cache personnalisé met en cache les répertoires que vous spécifiez dans le fichier buildspec. Ce mode est un bon choix si votre scénario de génération n'est pas adapté à un des deux autres modes de cache local. Si vous utilisez un cache personnalisé :
 - Seuls les répertoires peuvent être spécifiés pour la mise en cache. Vous ne pouvez pas spécifier de fichiers individuels.
 - Les liens symboliques sont utilisés pour référencer les répertoires mis en cache.
 - Les répertoires mis en cache sont liés à votre génération avant que ses sources de projet soient téléchargées. Les éléments mis en cache remplacent les éléments source s'ils portent le même nom. Les répertoires sont spécifiés à l'aide de chemins de cache dans le fichier buildspec. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).
 - Évitez les noms de répertoire identiques dans la source et dans le cache. Les répertoires mis en cache localement peuvent remplacer ou supprimer le contenu des répertoires du référentiel source portant le même nom.

Note

La mise en cache locale n'est pas prise en charge avec le type d'`LINUX_GPU_CONTAINER` environnement et le type de `BUILD_GENERAL1_2XLARGE` calcul. Pour plus d'informations, consultez [Modes et types de calcul de l'environnement de création](#).

Note

La mise en cache locale n'est pas prise en charge lorsque vous configurez CodeBuild pour fonctionner avec un VPC. Pour plus d'informations sur l'utilisation de VPC avec CodeBuild, consultez [Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Rubriques

- [Spécification d'une mise en cache locale \(CLI\)](#)
- [Spécification d'une mise en cache locale \(console\)](#)

- [Spécification d'une mise en cache locale \(AWS CloudFormation\)](#)

Vous pouvez utiliser la console AWS CLI, le SDK ou AWS CloudFormation pour spécifier un cache local.

Spécification d'une mise en cache locale (CLI)

Vous pouvez utiliser le paramètre `--cache` dans l'AWS CLI pour spécifier chacun des trois types de cache local.

- Pour spécifier un cache source :

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Pour spécifier un cache de couche Docker :

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

- Pour spécifier un cache personnalisé :

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

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

Spécification d'une mise en cache locale (console)

Vous spécifiez un cache dans la section Artifacts (Artefacts) de la console. Pour le type de cache, choisissez Amazon S3 ou Local. Si vous choisissez Local, choisissez une ou plusieurs des trois options de cache local.

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.

Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

Spécification d'une mise en cache locale (AWS CloudFormation)

Si vous utilisez AWS CloudFormation pour spécifier un cache local, sur la propriété `Cache`, pour `Type`, spécifiez `LOCAL`. L'exemple de code AWS CloudFormation au format YAML suivant spécifie les trois types de cache local. Vous pouvez spécifier n'importe quelle combinaison de types. Si vous utilisez un cache de couche Docker, sous `Environment`, vous devez définir `PrivilegedMode` sur `true` et `Type` sur `LINUX_CONTAINER`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: <service-role>
    Artifacts:
      Type: S3
      Location: <bucket-name>
      Name: myArtifact
      EncryptionDisabled: true
      OverrideArtifactName: true
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
      Certificate: <bucket/cert.zip>
      # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
      PrivilegedMode: true
    Source:
```

```
Type: GITHUB
Location: <github-location>
InsecureSsl: true
GitCloneDepth: 1
ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
  Type: LOCAL
  Modes: # You can specify one or more cache mode,
    - LOCAL_CUSTOM_CACHE
    - LOCAL_DOCKER_LAYER_CACHE
    - LOCAL_SOURCE_CACHE
```

Note

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

Pour plus d'informations, consultez [Création d'un projet de génération \(AWS CloudFormation\)](#).

Intégrez des déclencheurs dans AWS CodeBuild

Rubriques

- [Création de déclencheurs AWS CodeBuild](#)
- [Modification des déclencheurs AWS CodeBuild](#)

Création de déclencheurs AWS CodeBuild

Création de AWS CodeBuild déclencheurs (console)

Vous pouvez créer un déclencheur sur un projet pour planifier une génération une fois toutes les heures, tous les jours ou toutes les semaines. Vous pouvez également créer un déclencheur à l'aide d'une règle personnalisée avec une expression Amazon CloudWatch cron. Par exemple, à l'aide d'une expression cron, vous pouvez planifier une génération à une heure précise chaque jour de la semaine.

Note

Il n'est pas possible de démarrer une génération par lots à partir d'un déclencheur de génération, d'un EventBridge événement Amazon ou d'une AWS Step Functions tâche.

Pour créer un déclencheur

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération.
3. Sélectionnez le lien pour le projet de génération pour lequel vous voulez ajouter un déclencheur, puis choisissez l'onglet Build triggers (Déclencheurs de génération).

Note

Par défaut, les 100 projets de construction les plus récents sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

4. Choisissez Create trigger.
5. Saisissez un nom dans Trigger name (Nom du déclencheur).
6. Dans la liste déroulante Fréquence, choisissez la fréquence de votre déclencheur. Si vous voulez créer une fréquence à l'aide d'une expression cron, choisissez Custom (Personnalisé).
7. Spécifiez les paramètres de la fréquence de votre déclencheur. Vous pouvez saisir les tout premiers caractères de vos sélections dans la zone de texte pour filtrer les éléments du menu déroulant.

Note

Les heures et minutes de début sont basées sur zéro. La minute de début est un nombre compris entre zéro et 59. L'heure de début est un nombre compris entre zéro et 23. Par exemple, un déclencheur quotidien qui commence tous les jours à 12 h 15 a une heure de début de 12 heures et une minute de début de 15. Un déclencheur quotidien qui commence tous les jours à minuit a une heure de début de zéro et une minute de début

de zéro. Un déclencheur quotidien qui commence tous les jours à 23 h 59 a une heure de début de 23 heures et une minute de départ de 59.

| Fréquence | Paramètres requis | Détails |
|--------------|--|--|
| Par heure | Minute de début | Utilisez le menu déroulant Start minute (Minute de début). |
| Chaque jour | Minute de début Heure de début | Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début). |
| Hebdomadaire | Minute de début Heure de début Jour de début | Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début). Utilisez le menu déroulant Start day (Jour de début). |

| Fréquence | Paramètres requis | Détails |
|--------------|-------------------|---|
| Personnalisé | Expression cron | <p>Saisissez une expression cron dans Cron expression (Expression Cron).</p> <p>Une expression cron est composée de six champs obligatoires qui sont séparés par des espaces. Les champs spécifient une valeur de début pour la minute, l'heure, le jour du mois, le mois, le jour de la semaine et l'année. Vous pouvez utiliser des caractères génériques pour spécifier une plage, des valeurs supplémentaires, etc. Par exemple, l'expression cron 0 9 ? * MON-FRI * planifie une génération tous les jours de la semaine à 9 h 00. Pour plus d'informations, consultez la section Cron Expressions dans le guide de l'utilisateur Amazon CloudWatch Events.</p> |

8. Sélectionnez **Enable this trigger (Activer ce déclencheur)**.
9. (Facultatif) Développez **Advanced section (Section avancée)**. Dans **Source version (Version de la source)**, saisissez une version de votre code source.
 - Pour Amazon S3, entrez l'ID de version qui correspond à la version de l'artefact d'entrée que vous souhaitez créer. Si le champ **Version de la source** reste vide, la dernière version est utilisée.
 - Pour AWS CodeCommit, entrez un ID de validation. Si **Version de la source** est vide, l'ID de validation **HEAD** de la branche par défaut est utilisé.

- Pour GitHub ou GitHub Enterprise, saisissez un identifiant de validation, un identifiant de pull request, un nom de branche ou un nom de balise correspondant à la version du code source que vous souhaitez créer. Si vous spécifiez un ID de demande d'extraction, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si vous spécifiez un nom de branche, l'ID de validation HEAD de la branche est utilisé. Si Version source est vide, l'ID de validation HEAD de la branche par défaut est utilisé.
 - Pour Bitbucket, entrez un ID de validation, un nom de branche ou un nom de balise qui correspond à la version du code source à générer. Si vous spécifiez un nom de branche, l'ID de validation HEAD de la branche est utilisé. Si Version source est vide, l'ID de validation HEAD de la branche par défaut est utilisé.
10. (Facultatif) Spécifiez un délai d'expiration compris entre 5 minutes et 480 minutes (8 heures). Cette valeur spécifie pendant combien de temps AWS CodeBuild tente une génération avant de s'arrêter. Si les Hours (heures) et Minutes sont laissées vides, la valeur de délai d'expiration par défaut spécifiée dans le projet est utilisée.
 11. Choisissez Create trigger.

Création de AWS CodeBuild déclencheurs par programmation

CodeBuild utilise les EventBridge règles d'Amazon pour les déclencheurs de création. Vous pouvez utiliser l' EventBridge API pour créer par programmation des déclencheurs de build pour vos CodeBuild projets. Consultez le [manuel Amazon EventBridge API Reference](#) pour plus d'informations.

Modification des déclencheurs AWS CodeBuild

Modifier les AWS CodeBuild déclencheurs (console)

Vous pouvez modifier un déclencheur sur un projet pour planifier une génération une fois toutes les heures, tous les jours ou toutes les semaines. Vous pouvez également modifier un déclencheur pour utiliser une règle personnalisée avec une expression Amazon CloudWatch Cron. Par exemple, à l'aide d'une expression cron, vous pouvez planifier une génération à une heure précise chaque jour de la semaine. Pour plus d'informations sur la création d'un déclencheur, consultez [Création de déclencheurs AWS CodeBuild](#).

Pour modifier un déclencheur


1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).

2. Dans le volet de navigation, choisissez Projets de génération.
3. Sélectionnez le lien du projet de génération à modifier, puis choisissez l'onglet Build triggers (Déclencheurs de génération).

 Note

Par défaut, les 100 projets de construction les plus récents sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

4. Sélectionnez la case d'option en regard du déclencheur à modifier, puis choisissez Edit (Modifier).
5. Dans la liste déroulante Fréquence, choisissez la fréquence de votre déclencheur. Si vous voulez créer une fréquence à l'aide d'une expression cron, choisissez Custom (Personnalisé).
6. Spécifiez les paramètres de la fréquence de votre déclencheur. Vous pouvez saisir les tout premiers caractères de vos sélections dans la zone de texte pour filtrer les éléments du menu déroulant.

 Note


Les heures et minutes de début sont basées sur zéro. La minute de début est un nombre compris entre zéro et 59. L'heure de début est un nombre compris entre zéro et 23. Par exemple, un déclencheur quotidien qui commence tous les jours à 12 h 15 a une heure de début de 12 heures et une minute de début de 15. Un déclencheur quotidien qui commence tous les jours à minuit a une heure de début de zéro et une minute de début de zéro. Un déclencheur quotidien qui commence tous les jours à 23 h 59 a une heure de début de 23 heures et une minute de départ de 59.

| Fréquence | Paramètres requis | Détails |
|-----------|-------------------|--|
| Par heure | Minute de début | Utilisez le menu déroulant Start minute (Minute de début). |

| Fréquence | Paramètres requis | Détails |
|--------------|--|--|
| Chaque jour | Minute de début Heure de début | Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début). |
| Hebdomadaire | Minute de début Heure de début Jour de début | Utilisez le menu déroulant Start minute (Minute de début). Utilisez le menu déroulant Start hour (Heure de début). Utilisez le menu déroulant Start day (Jour de début). |

| Fréquence | Paramètres requis | Détails |
|--------------|-------------------|--|
| Personnalisé | Expression cron | Saisissez une expression cron dans Cron expression (Expression Cron). Une expression cron est composée de six champs obligatoires qui sont séparés par des espaces. Les champs spécifient une valeur de début pour la minute, l'heure, le jour du mois, le mois, le jour de la semaine et l'année. Vous pouvez utiliser des caractères génériques pour spécifier une plage, des valeurs supplémentaires, etc. Par exemple, l'expression cron 0 9 ? * MON-FRI * planifie une génération tous les jours de la semaine à 9 h 00. Pour plus d'informations, consultez la section Cron Expressions dans le guide de l'utilisateur Amazon CloudWatch Events . |

7. Sélectionnez **Enable this trigger (Activer ce déclencheur)**.

 Note

Vous pouvez utiliser la CloudWatch console Amazon à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/) pour modifier la version de la source, le délai d'expiration et d'autres options qui ne sont pas disponibles dans AWS CodeBuild.

Modifier les AWS CodeBuild déclencheurs par programmation

CodeBuild utilise les EventBridge règles d'Amazon pour les déclencheurs de création. Vous pouvez utiliser l' EventBridge API pour modifier par programmation les déclencheurs de génération de vos CodeBuild projets. Consultez le [manuel Amazon EventBridge API Reference](#) pour plus d'informations.

Utilisation de webhooks avec AWS CodeBuild

AWS CodeBuild prend en charge l'intégration de webhook avec GitHub GitHub Enterprise Server et Bitbucket.

Rubriques

- [Bonnes pratiques d'utilisation des webhooks avec AWS CodeBuild](#)
- [Événements du webhook Bitbucket](#)
- [GitHub événements webhook](#)

Bonnes pratiques d'utilisation des webhooks avec AWS CodeBuild

Pour les projets qui utilisent des référentiels publics pour configurer des webhooks, nous recommandons les options suivantes :

ACTOR_ACCOUNT_IDFiltres de configuration

Ajoutez des ACTOR_ACCOUNT_ID filtres aux groupes de filtres Webhook de votre projet pour spécifier quels utilisateurs peuvent déclencher une génération. Chaque événement webhook CodeBuild livré à est accompagné d'informations sur l'expéditeur qui spécifient l'identifiant de l'acteur. CodeBuild filtrera les webhooks en fonction du modèle d'expression régulière fourni dans les filtres. Vous pouvez spécifier les utilisateurs spécifiques autorisés à déclencher des builds avec ce filtre. Pour plus d'informations, consultez [GitHub événements webhook](#) et [Événements du webhook Bitbucket](#).

FILE_PATHFiltres de configuration

Ajoutez des FILE_PATH filtres aux groupes de filtres Webhook de votre projet pour inclure ou exclure les fichiers susceptibles de déclencher une génération en cas de modification. Par exemple, vous pouvez refuser les demandes de compilation visant à modifier le `buildspec.yml` fichier à l'aide d'un modèle d'expression régulière tel que `^buildspec.yml$`, avec la

`excludeMatchedPattern` propriété. Pour plus d'informations, consultez [GitHub événements webhook](#) et [Événements du webhook Bitbucket](#).

Limitez les autorisations pour votre rôle IAM de build

Les builds déclenchés par un webhook utilisent le rôle de service IAM spécifié dans le projet. Nous vous recommandons de définir les autorisations du rôle de service sur l'ensemble minimal d'autorisations requis pour exécuter la génération. Par exemple, dans un scénario de test et de déploiement, créez un projet pour les tests et un autre pour le déploiement. Le projet de test accepte les versions de webhook à partir du référentiel, mais ne fournit aucune autorisation d'écriture sur vos ressources. Le projet de déploiement fournit des autorisations d'écriture à vos ressources, et le filtre Webhook est configuré pour autoriser uniquement les utilisateurs de confiance à déclencher des builds.

Utilisez une spécification de construction en ligne ou stockée sur Amazon S3

Si vous définissez votre buildspec en ligne dans le projet lui-même, ou si vous stockez le fichier buildspec dans un compartiment Amazon S3, le fichier buildspec n'est visible que par le propriétaire du projet. Cela empêche les pull requests de modifier le code du fichier buildspec et de déclencher des builds indésirables. Pour plus d'informations, consultez [ProjectSource.buildspec](#) dans la référence de l'API. CodeBuild

Événements du webhook Bitbucket

Vous pouvez utiliser des groupes de filtres webhook pour spécifier quels événements webhook Bitbucket déclenchent une génération. Par exemple, vous pouvez spécifier qu'une construction n'est déclenchée que pour les modifications apportées à des branches spécifiques.

Vous pouvez créer un ou plusieurs des groupes de filtres webhook pour spécifier les événements webhook qui déclenchent une génération. Une génération est déclenchée si un groupe de filtres est évalué comme vrai, ce qui se produit lorsque tous les filtres du groupe sont évalués comme vrais. Lorsque vous créez un groupe de filtres, vous spécifiez :


Un événement

Pour Bitbucket, vous pouvez choisir un ou plusieurs des événements suivants :

- PUSH
- PULL_REQUEST_CREATED
- PULL_REQUEST_UPDATED

- `PULL_REQUEST_MERGED`
- `PULL_REQUEST_CLOSED`

Le type d'événement du webhook est situé dans son en-tête dans le champ `X-Event-Key`. Le tableau suivant présente le mappage des valeurs d'en-tête `X-Event-Key` aux types d'événements.

 Note

Vous devez activer l'événement `merged` dans le paramètre de votre webhook Bitbucket si vous créez un groupe de filtres de webhook qui utilise le type d'événement `PULL_REQUEST_MERGED`. Vous devez également activer l'événement `declined` dans vos paramètres de webhook Bitbucket si vous créez un groupe de filtres de webhook utilisant le `PULL_REQUEST_CLOSED` type d'événement.

| Valeur d'en-tête <code>X-Event-Key</code> | Type d'événement |
|---|-----------------------------------|
| <code>repo:push</code> | <code>PUSH</code> |
| <code>pullrequest:created</code> | <code>PULL_REQUEST_CREATED</code> |
| <code>pullrequest:updated</code> | <code>PULL_REQUEST_UPDATED</code> |
| <code>pullrequest:fulfilled</code> | <code>PULL_REQUEST_MERGED</code> |
| <code>pullrequest:rejected</code> | <code>PULL_REQUEST_CLOSED</code> |

En `PULL_REQUEST_MERGED` effet, si une pull request est fusionnée avec la stratégie squash et que la branche pull request est fermée, le commit de la pull request d'origine n'existe plus. Dans ce cas, la variable d'environnement `CODEBUILD_WEBHOOK_MERGE_COMMIT` contient l'identifiant du commit de fusion écrasé.

Un ou plusieurs filtres optionnels

Utilisez une expression régulière pour spécifier un filtre. Pour qu'un événement déclenche une génération, chaque filtre du groupe qui lui est associé doit avoir la valeur `true`.

ACTOR_ACCOUNT_ID(ACTOR_ID dans la console)

Un événement webhook déclenche une compilation lorsqu'un identifiant de compte Bitbucket correspond au modèle d'expression régulière. Cette valeur est disponible dans la propriété `account_id` de l'objet `actor` dans la charge utile de filtre du webhook.

HEAD_REF

Un événement webhook déclenche une construction lorsque la référence principale correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name` et `refs/tags/tag-name`). Un filtre `HEAD_REF` évalue le nom de la référence Git de la branche ou de la balise. Le nom de la branche ou de la balise est situé dans le champ `name` de l'objet `new` dans l'objet `push` de la charge utile du webhook. Pour des événements de demande d'extraction, le nom de la branche est situé dans le champ `name` de l'objet `branch` de l'objet `source` dans la charge utile du webhook.

BASE_REF

Un événement webhook déclenche une construction lorsque la référence de base correspond au modèle d'expression régulière. Un filtre `BASE_REF` fonctionne avec des événements de demande d'extraction uniquement (par exemple, `refs/heads/branch-name`). Un filtre `BASE_REF` évalue le nom de la référence Git de la branche. Le nom de la branche est situé dans le champ `name` de l'objet `branch` dans l'objet `destination` de la charge utile du webhook.

FILE_PATH

Un webhook déclenche une compilation lorsque le chemin d'un fichier modifié correspond au modèle d'expression régulière.

COMMIT_MESSAGE

Un webhook déclenche une compilation lorsque le message de validation principal correspond au modèle d'expression régulière.

Note

Vous pouvez trouver la charge utile du webhook dans les paramètres webhook de votre référentiel Bitbucket.

Rubriques

- [Filtrage des événements webhook BitBucket \(console\)](#)
- [Filtrage des événements webhook BitBucket \(kit SDK\)](#)
- [Filtrage des événements webhook Bitbucket \(AWS CloudFormation\)](#)

Filtrage des événements webhook BitBucket (console)

Pour utiliser le pour filtrer AWS Management Console les événements du webhook, procédez comme suit :

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Add filter group (Ajouter un groupe de filtres) pour ajouter un autre groupe de filtres.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API AWS CodeBuild .

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour des demandes d'extraction uniquement :

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ×PULL_REQUEST_UPDATED ×PULL_REQUEST_MERGED ×PULL_REQUEST_CLOSED ×

▶ **Start a build under these conditions - optional**

▶ **Don't start a build under these conditions - optional**

Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur true :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/branch1!`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/branch1$`.

Webhook event filter group 1

Event type
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

| ACTOR_ID - optional | HEAD_REF - optional | BASE_REF - optional | FILE_PATH - optional |
|----------------------|--|---|----------------------|
| <input type="text"/> | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text" value="^refs/heads/main\$"/> | <input type="text"/> |

COMMIT_MESSAGE - optional

► **Don't start a build under these conditions**

Webhook event filter group 2

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.

▼ **Start a build under these conditions**

| ACTOR_ID - optional | HEAD_REF - optional | BASE_REF - optional | FILE_PATH - optional |
|----------------------|--|----------------------|----------------------|
| <input type="text"/> | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text"/> | <input type="text"/> |

COMMIT_MESSAGE - optional

► **Don't start a build under these conditions**

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour toutes les demandes à l'exception des événements de balise.

Filter group 1 Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕PULL_REQUEST_CREATED ✕PULL_REQUEST_UPDATED ✕

PULL_REQUEST_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

HEAD_REF

Pattern

^refs/tags/.*

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

 COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

 COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur Bitbucket qui ne dispose pas d'un ID de compte correspondant à l'expression régulière `actor-account-id`.

Note

Pour plus d'informations sur la façon de trouver votre ID de compte Bitbucket, consultez <https://api.bitbucket.org/2.0/users/nom-utilisateur>, où *nom-utilisateur* est votre nom d'utilisateur Bitbucket.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 2

Type

Pattern

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour un événement de transmission lorsque le message de validation principal correspond à l'expression régulière `\[CodeBuild\]`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

▶ Don't start a build under these conditions

Filtrage des événements webhook BitBucket (kit SDK)

Pour utiliser le AWS CodeBuild SDK afin de filtrer les événements Webhook, utilisez le `filterGroups` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Pour créer un filtre webhook qui déclenche une génération pour des demandes d'extraction uniquement, insérez les éléments suivants dans la syntaxe de demande :

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED,
PULL_REQUEST_CLOSED"
    }
  ]
]
```

Pour créer un filtre webhook qui déclenche une génération pour des branches spécifiées uniquement, utilisez le paramètre `pattern` pour spécifier une expression régulière pour filtrer les noms de branche. Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur true :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` et des références principales correspondant à `^refs/heads/myBranch$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Vous pouvez utiliser le paramètre `excludeMatchedPattern` pour spécifier les événements qui ne déclenchent pas une génération. Dans cet exemple, une génération est déclenchée pour toutes les demandes à l'exception des événements de balise.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```



```

    "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
  },
  {
    "type": "HEAD_REF",
    "pattern": "^refs/tags/.*",
    "excludeMatchedPattern": true
  }
]
]

```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur Bitbucket avec l'ID de compte `actor-account-id`.

Note

Pour plus d'informations sur la façon de trouver votre ID de compte Bitbucket, consultez <https://api.bitbucket.org/2.0/users/nom-utilisateur>, où *nom-utilisateur* est votre nom d'utilisateur Bitbucket.

```

"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]

```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière dans l'argument `pattern` changent. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

```

"filterGroups": [

```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "FILE_PATH",
    "pattern": "^buildspec.*"
  }
]
```

Dans cet exemple, le groupe de filtres indique qu'une compilation est déclenchée uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque le message de validation principal correspond à l'expression régulière de l'argument de schéma. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque le message de validation principal de l'événement de transmission correspond à l'expression régulière `\[CodeBuild\]`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "COMMIT_MESSAGE",
```

```
    "pattern": "\\[CodeBuild\\]"
  }
]
]
```

Filtrage des événements webhook Bitbucket (AWS CloudFormation)

Pour utiliser un AWS CloudFormation modèle afin de filtrer les événements Webhook, utilisez la `FilterGroups` propriété du AWS CodeBuild projet. La partie au format YAML suivante d'un modèle AWS CloudFormation crée deux groupes de filtres. Ensemble, ils déclenchent une génération lorsque l'un des deux ou les deux prennent la valeur `true` :

- Le premier groupe de filtres spécifie des demandes d'extraction créées ou mises à jour sur des branches avec des noms de référence Git correspondant à l'expression régulière `^refs/heads/main$` par un utilisateur Bitbucket n'ayant pas d'ID de compte 12345.
- Le deuxième groupe de filtres spécifie des demandes de transmission créées sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\\[CodeBuild\\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
```

```
    Pattern: ^refs/heads/main$
    ExcludeMatchedPattern: false
  - Type: ACTOR_ACCOUNT_ID
    Pattern: 12345
    ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
  - Type: HEAD_REF
    Pattern: ^refs/heads/.+
  - Type: FILE_PATH
    Pattern: README
    ExcludeMatchedPattern: true
- - Type: EVENT
    Pattern: PUSH
  - Type: COMMIT_MESSAGE
    Pattern: \[CodeBuild\]
  - Type: FILE_PATH
    Pattern: ^src/.+|^test/.+
```

GitHub événements webhook

Vous pouvez utiliser des groupes de filtres Webhook pour spécifier quels événements GitHub Webhook déclenchent une compilation. Par exemple, vous pouvez spécifier qu'une construction n'est déclenchée que pour les modifications apportées à des branches spécifiques.

Vous pouvez créer un ou plusieurs des groupes de filtres webhook pour spécifier les événements webhook qui déclenchent une génération. Une génération est déclenchée si un groupe de filtres est évalué comme vrai, ce qui se produit lorsque tous les filtres du groupe sont évalués comme vrais. Lorsque vous créez un groupe de filtres, vous spécifiez :

Un événement


En GitHub effet, vous pouvez choisir un ou plusieurs des événements suivants :

PUSHPULL_REQUEST_CREATED,

PULL_REQUEST_UPDATED,PULL_REQUEST_REOPENED,PULL_REQUEST_MERGED,PULL_REQUEST_CLOSED

etPRERELEASED. Le type d'événement du webhook est situé dans l'en-tête X-GitHub-Event de la charge utile du webhook. Dans l'en-tête X-GitHub-Event, vous pourriez voir pull_request ou push. Pour un événement de demande d'extraction, le type se situe dans le champ action de la charge utile d'événement du webhook. Le tableau suivant présente le mappage des valeurs de l'en-tête X-GitHub-Event et des valeurs du champ action de la charge utile de la demande d'extraction du webhook aux types d'événements disponibles.

| Valeur d'en-tête X-GitHub-Event | Valeur action de la charge utile de l'événement du webhook | Type d'événement |
|--|---|-----------------------|
| pull_request | opened | PULL_REQUEST_CREATED |
| pull_request | reopened | PULL_REQUEST_REOPENED |
| pull_request | synchronize | PULL_REQUEST_UPDATED |
| pull_request | closed et le champ merged est true | PULL_REQUEST_MERGED |
| pull_request | closed et le champ merged est false | PULL_REQUEST_CLOSED |
| push | N/A | PUSH |
| release | publié | RELEASED |
| release | pré-publié | PRERELEASED |

 Note

Le type d'`PULL_REQUEST_REOPENED` événement ne peut être utilisé qu'avec un GitHub serveur GitHub d'entreprise. Le type `PRERELEASED` d'événement `RELEASED` et ne peut être utilisé GitHub qu'avec.

Un ou plusieurs filtres optionnels

Utilisez une expression régulière pour spécifier un filtre. Pour qu'un événement déclenche une génération, chaque filtre du groupe qui lui est associé doit avoir la valeur true.

`ACTOR_ACCOUNT_ID(ACTOR_ID dans la console)`

Un événement webhook déclenche une compilation lorsqu'un identifiant de compte GitHub ou un identifiant de compte GitHub Enterprise Server correspond au modèle d'expression

régulière. Cette valeur se situe dans la propriété `id` de l'objet `sender` dans la charge utile du `webhook`.

HEAD_REF

Un événement `webhook` déclenche une construction lorsque la référence principale correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name` ou `refs/tags/tag-name`). Pour un événement d'extraction, le nom de référence est situé dans la propriété `ref` de la charge utile du `webhook`. Pour les événements de demande d'extraction, le nom de la branche est situé dans la propriété `ref` de l'objet `head` dans la charge utile du `webhook`.

BASE_REF

Un événement `webhook` déclenche une construction lorsque la référence de base correspond au modèle d'expression régulière (par exemple, `refs/heads/branch-name`). Un filtre `BASE_REF` ne peut être utilisé qu'avec des événements de demande d'extraction. Le nom de la branche se situe dans la propriété `ref` de l'objet `base` dans la charge utile du `webhook`.

FILE_PATH

Un `webhook` déclenche une compilation lorsque le chemin d'un fichier modifié correspond au modèle d'expressions régulières. Un `FILE_PATH` filtre peut être utilisé avec les événements de requêtes GitHub `push` et `pull` et les événements `push` GitHub d'Enterprise Server. Il ne peut pas être utilisé avec les événements de `pull request` d' GitHubEnterprise Server.

COMMIT_MESSAGE

Un `webhook` déclenche une compilation lorsque le message de validation principal correspond au modèle d'expression régulière. Un `COMMIT_MESSAGE` filtre peut être utilisé avec les événements de requêtes GitHub `push` et `pull` et les événements `push` GitHub d'Enterprise Server. Il ne peut pas être utilisé avec les événements de `pull request` d' GitHubEnterprise Server.

TAG_NAME

Un `webhook` déclenche une compilation lorsque le nom de balise de la version correspond au modèle d'expression régulière. Un `TAG_NAME` filtre peut être utilisé avec les événements de demande GitHub publiés et prépubliés.

RELEASE_NAME

Un webhook déclenche une compilation lorsque le nom de la version correspond au modèle d'expression régulière. Un RELEASE_NAME filtre peut être utilisé avec les événements de demande GitHub publiés et prépubliés.

Note

Vous pouvez trouver la charge utile du webhook dans les paramètres du webhook de votre dépôt. GitHub

Rubriques

- [Filtrer les événements du GitHub webhook \(console\)](#)
- [Filtrer les événements du GitHub webhook \(SDK\)](#)
- [Filtrer les événements du GitHub webhook \(AWS CloudFormation\)](#)

Filtrer les événements du GitHub webhook (console)

Dans Événements webhook de la source principale, sélectionnez ce qui suit. Cette section n'est disponible que lorsque vous avez choisi Repository dans mon GitHub compte pour le référentiel source.

1. Sélectionnez Rebuild every time a code change is pushed to this repository (Reconstruire à chaque fois qu'une modification de code est transférée dans ce référentiel) lorsque vous créez votre projet.
2. Dans Event type (Type d'événement), choisissez un ou plusieurs événements.
3. Pour filtrer le moment où un événement déclenche une génération, sous Start a build under these conditions (Lancer une génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
4. Pour filtrer le moment où un événement n'est pas déclenché, sous Don't start a build under these conditions (Ne pas lancer de génération dans ces conditions), ajoutez un ou plusieurs filtres facultatifs.
5. Choisissez Ajouter un groupe de filtres pour ajouter un autre groupe de filtres, si nécessaire.

Pour plus d'informations, veuillez consulter les sections [Création d'un projet de génération \(console\)](#) et [WebhookFilter](#) (français non garanti) de la Référence d'API AWS CodeBuild .

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour des demandes d'extraction uniquement :

Filter group 1 Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

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

Dans cet exemple de deux groupes de filtres webhook, une génération est déclenchée lorsque l'un des deux ou les deux prennent la valeur true :

- Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/main$` et des références principales qui correspondent à `^refs/heads/branch1$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_REOPENED ✕

▼ 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

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour toutes les demandes à l'exception des événements de balise.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)[PULL_REQUEST_CREATED X](#)[PULL_REQUEST_UPDATED X](#)[PULL_REQUEST_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*

[Add filter](#)

Filter 1

Type

Pattern

Dans cet exemple, un groupe de filtres webhook déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une compilation uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Dans cet exemple, un groupe de filtres Webhook déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur spécifié GitHub ou par un utilisateur d' GitHub Enterprise Server dont l'ID de compte correspond à l'expression `actor-account-id` régulière.

Note

Pour savoir comment trouver l'identifiant de votre GitHub compte, consultez [https://api.github.com/users/ nom d'utilisateur](https://api.github.com/users/nom d'utilisateur), où *nom d'utilisateur* est votre nom GitHub d'utilisateur.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_REOPENED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 2

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Dans cet exemple, un groupe de filtres webhook déclenche une génération pour un événement de transmission lorsque le message de validation principal correspond à l'expression régulière `\[CodeBuild\]`.

Webhook event filter 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

Filtrer les événements du GitHub webhook (SDK)

Pour utiliser le AWS CodeBuild SDK afin de filtrer les événements Webhook, utilisez le `filterGroups` champ dans la syntaxe de demande des méthodes `CreateWebhook` ou de `UpdateWebhookAPI`. Pour plus d'informations, consultez [WebhookFilter](#) dans la Référence d'API CodeBuild .

Pour créer un filtre webhook qui déclenche une génération pour des demandes d'extraction uniquement, insérez les éléments suivants dans la syntaxe de demande :

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    }
  ]
]
```

Pour créer un filtre webhook qui déclenche une génération pour des branches spécifiées uniquement, utilisez le paramètre `pattern` pour spécifier une expression régulière pour filtrer les noms de branche. Dans cet exemple de deux groupes de filtres, une génération est déclenchée lorsqu'un des deux prend la valeur `true` :

- Le premier groupe de filtres spécifie les demandes d'extraction créées, mises à jour ou rouvertes sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/main$` et des références principales qui correspondent à `^refs/heads/myBranch$`.
- Le deuxième groupe de filtres spécifie des demandes de transmission sur des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Vous pouvez utiliser le paramètre `excludeMatchedPattern` pour spécifier les événements qui ne déclenchent pas une génération. Dans cet exemple, une génération est déclenchée pour toutes les demandes sauf les événements de balise.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
  },
  {
    "type": "HEAD_REF",
    "pattern": "^refs/tags/.*",
    "excludeMatchedPattern": true
  }
]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière dans l'argument `pattern` changent. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque les fichiers dont les noms correspondent à l'expression régulière `^buildspec.*` changent.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

Dans cet exemple, le groupe de filtres indique qu'une compilation est déclenchée uniquement lorsque des fichiers sont modifiés dans `src` ou `test` des dossiers.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
```

```
        "type": "FILE_PATH",
        "pattern": "^src/.+|^test/.+"
    }
]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsqu'une modification est apportée par un utilisateur spécifié GitHub ou par un utilisateur GitHub Enterprise Server doté d'un identifiant de compte `actor-account-id`.

Note

Pour savoir comment trouver l'identifiant de votre GitHub compte, consultez https://api.github.com/users/nom_d'utilisateur, où *nom d'utilisateur* est votre nom GitHub d'utilisateur.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

Vous pouvez créer un filtre qui déclenche une génération uniquement lorsque le message de validation principal correspond à l'expression régulière de l'argument de schéma. Dans cet exemple, le groupe de filtres spécifie qu'une génération est déclenchée uniquement lorsque le message de validation principal de l'événement de transmission correspond à l'expression régulière `\[CodeBuild\]`.

```
"filterGroups": [
  [
    {
```



```

        "type": "EVENT",
        "pattern": "PUSH"
    },
    {
        "type": "COMMIT_MESSAGE",
        "pattern": "\\[CodeBuild\\]"
    }
]
]

```

Filtrer les événements du GitHub webhook ()AWS CloudFormation

Pour utiliser un AWS CloudFormation modèle afin de filtrer les événements Webhook, utilisez la `FilterGroups` propriété du AWS CodeBuild projet. La partie au format YAML suivante d'un modèle AWS CloudFormation crée deux groupes de filtres. Ensemble, ils déclenchent une génération lorsque l'un des deux ou les deux prennent la valeur `true` :

- Le premier groupe de filtres indique que les pull requests sont créées ou mises à jour sur les branches dont les noms de référence Git correspondent à l'expression régulière `^refs/heads/main$` d'un GitHub utilisateur qui n'a pas d'identifiant de compte `12345`.
- Le deuxième groupe de filtres spécifie les demandes de transmission créées sur des fichiers dont les noms correspondent à l'expression régulière `READ_ME` dans des branches avec des noms de référence Git qui correspondent à l'expression régulière `^refs/heads/.*`.
- Le troisième groupe de filtres spécifie une demande de transmission avec un message de validation principal correspondant à l'expression régulière `\\[CodeBuild\\]`.

```

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/.+
```

Modification des paramètres d'un projet de génération dans AWS CodeBuild

Vous pouvez utiliser la console AWS CodeBuild, l'AWS CLI, ou des kits SDK AWS pour modifier les paramètres d'un projet de génération.

Si vous ajoutez des rapports de test à un projet de génération, assurez-vous que votre rôle IAM dispose des autorisations décrites à la section [Utilisation des autorisations de rapport de test](#).

Rubriques

- [Modification des paramètres d'un projet de génération \(console\)](#)
- [Modification des paramètres d'un projet de génération \(AWS CLI\)](#)
- [Modification des paramètres d'un projet de génération \(kits SDK AWS\)](#)

Modification des paramètres d'un projet de génération (console)

Pour modifier les paramètres d'un projet de génération, effectuez la procédure suivante :

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :
 - Cliquez sur le lien du projet de génération à modifier, puis choisissez Détails de génération.
 - Cliquez sur le bouton en regard du projet de génération à modifier, choisissez View details (Afficher les détails), puis Détails de génération.

Vous pouvez modifier les sections suivantes :

Sections

- [Configuration du projet](#)
- [Source](#)
- [Environnement](#)
- [Spécifications de construction](#)
- [Configuration par lots](#)
- [Artefacts](#)
- [Journaux](#)

Configuration du projet

Dans la section Configuration du projet, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes.

Description

Entrez une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.

Construire un badge

Sélectionnez Enable build badge (Activer le badge de génération) pour rendre le statut de génération de votre projet visible et intégrable. Pour plus d'informations, consultez [Exemple de badges de génération](#).

Note

Le badge de construction ne s'applique pas si votre fournisseur source est Amazon S3.

Activer la limite de génération simultanée

Si vous souhaitez limiter le nombre de builds simultanés pour ce projet, effectuez les opérations suivantes :

1. Sélectionnez Restreindre le nombre de versions simultanées que ce projet peut démarrer.
2. Dans Limite de génération simultanée, entrez le nombre maximum de versions simultanées autorisées pour ce projet. Cette limite ne peut pas être supérieure à la limite de création simultanée définie pour le compte. Si vous essayez de saisir un nombre supérieur à la limite du compte, un message d'erreur s'affiche.

Les nouvelles générations ne sont démarrées que si le nombre actuel de générations est inférieur ou égal à cette limite. Si le nombre actuel de générations atteint cette limite, les nouvelles générations sont limitées et ne sont pas exécutées.

Permettre l'accès public aux builds

Pour que les résultats de génération de votre projet soient accessibles au public, y compris aux utilisateurs n'ayant pas accès à un AWS compte, sélectionnez Activer l'accès public aux versions et confirmez que vous souhaitez rendre les résultats de génération publics. Les propriétés suivantes sont utilisées pour les projets de construction publics :

Rôle du service public de construction

Sélectionnez Nouveau rôle de service si vous souhaitez CodeBuild créer un nouveau rôle de service pour vous, ou Rôle de service existant si vous souhaitez utiliser un rôle de service existant.

Le rôle de service de construction public CodeBuild permet de lire les CloudWatch journaux et de télécharger les artefacts Amazon S3 pour les versions du projet. Cela est nécessaire pour mettre les journaux de construction et les artefacts du projet à la disposition du public.

Rôle de service

Entrez le nom du nouveau rôle de service ou d'un rôle de service existant.

Pour que les résultats de compilation de votre projet soient privés, décochez Activer l'accès public aux builds.

Pour plus d'informations, consultez [Projets de build public dans AWS CodeBuild](#).

Warning

Les points suivants doivent être pris en compte lorsque vous publiez les résultats de construction de votre projet :

- Tous les résultats de construction, les journaux et les artefacts d'un projet, y compris les versions exécutées lorsque le projet était privé, sont accessibles au public.
- Tous les journaux de construction et les artefacts sont accessibles au public. Les variables d'environnement, le code source et d'autres informations sensibles peuvent avoir été générés dans les journaux de construction et les artefacts. Vous devez faire attention aux informations qui sont affichées dans les journaux de construction. Voici quelques bonnes pratiques :
 - Ne stockez pas de valeurs sensibles, en particulier les identifiants de clé AWS d'accès et les clés d'accès secrètes, dans les variables d'environnement. Nous vous recommandons d'utiliser un magasin de paramètres Amazon EC2 Systems Manager AWS Secrets Manager ou de stocker des valeurs sensibles.
 - Suivez cette [Bonnes pratiques d'utilisation des webhooks](#) procédure pour limiter les entités qui peuvent déclencher une construction, et ne stockez pas la spécification de construction dans le projet lui-même, afin de garantir que vos webhooks sont aussi sécurisés que possible.
 - Un utilisateur malveillant peut utiliser des versions publiques pour distribuer des artefacts malveillants. Nous recommandons aux administrateurs de projet de passer en revue toutes les pull requests afin de vérifier qu'il s'agit d'une modification légitime. Nous vous recommandons également de valider tous les artefacts à l'aide de leurs checksums afin de vous assurer que les bons artefacts sont téléchargés.

Informations supplémentaires

Pour les balises, entrez le nom et la valeur de toutes les balises que vous souhaitez que les AWS services d'assistance utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.

Source

Dans la section Source, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Fournisseur de source

Choisissez le type de fournisseur de code source. Utilisez les listes suivantes pour effectuer des sélections adaptées à votre fournisseur de source :

Note

CodeBuild ne prend pas en charge Bitbucket Server.

Amazon S3

Compartiment

Choisissez le nom du compartiment d'entrée contenant le code source.

Clé d'objet S3 ou dossier S3

Entrez le nom du fichier ZIP ou le chemin d'accès au dossier contenant le code source. Entrez une barre oblique (/) pour tout télécharger dans le compartiment S3.

Version source

Entrez l'ID de version de l'objet qui représente la version de votre fichier d'entrée. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).


CodeCommit

Référentiel

Choisissez le référentiel que vous souhaitez utiliser.

Type de référence

Choisissez Branch, Git tag ou Commit ID pour spécifier la version de votre code source. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez de créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Bitbucket


Référentiel

Choisissez Connect using OAuth ou Connect with a Bitbucket app password, puis suivez les instructions pour vous connecter (ou vous reconnecter) à Bitbucket.

Choisissez un dépôt public ou un dépôt dans votre compte.

Version source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, veuillez consulter la page [Exemple de version source avec AWS CodeBuild](#).

 Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le name paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

Pour l'URL cible, entrez la valeur à utiliser pour le url paramètre dans le statut de validation de Bitbucket. Pour plus d'informations, voir [build](#) dans la documentation de l'API Bitbucket.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [Événements du webhook Bitbucket](#).

GitHub

Référentiel

Choisissez **Connect using OAuth** ou **Connect with a GitHub personal access token**, puis suivez les instructions pour vous connecter (ou vous reconnecter) GitHub et autoriser l'accès à AWS CodeBuild.

Choisissez un dépôt public ou un dépôt dans votre compte.

Version source

Entrez une branche, un ID de validation, une balise ou une référence et un ID de validation. Pour de plus amples informations, veuillez consulter la page [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que `811dd1ba1aba14473856cee38308caed7190c0d` ou `5392f7`. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez **Git clone depth (Profondeur du clone Git)** pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez **Full**.

Sous-modules Git

Sélectionnez **Use Git submodules (Utiliser les sous-modules Git)** si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez **Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds** si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose

pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le `context` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

GitHub Enterprise Server

GitHub Jeton d'accès personnel d'entreprise

Consultez [GitHub Exemple de serveur d'entreprise](#) pour savoir comment copier un jeton d'accès personnel dans votre presse-papiers. Collez le jeton dans la zone de texte, puis choisissez Save Token (Enregistrer le jeton).

Note

Vous n'avez besoin de saisir et d'enregistrer le jeton d'accès personnel qu'une seule fois. CodeBuild utilise ce jeton dans tous les futurs projets.

Version source

Entrez une pull request, une branche, un identifiant de validation, une balise ou une référence et un identifiant de validation. Pour plus d'informations, consultez [Exemple de version source avec AWS CodeBuild](#).

Note

Nous vous recommandons de choisir des noms de branche Git qui ne ressemblent pas à des identifiants de validation, tels que 811dd1ba1aba14473856cee38308caed7190c0d ou 5392f7. Cela vous permet d'éviter les collisions entre Git Checkout et les validations réelles.

Profondeur du clone Git

Choisissez Git clone depth (Profondeur du clone Git) pour créer un clone superficiel avec un historique tronqué au nombre de validations spécifié. Si vous souhaitez un clone complet, choisissez Full.

Sous-modules Git

Sélectionnez Use Git submodules (Utiliser les sous-modules Git) si vous souhaitez inclure les sous-modules Git dans votre référentiel.

Statut de la génération

Sélectionnez Signaler les statuts de construction au fournisseur source au début et à la fin de vos builds si vous souhaitez que l'état du début et de la fin de votre build soit signalé à votre fournisseur source.

Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour plus d'informations, consultez [Accès au fournisseur de source](#).

Pour le contexte de statut, entrez la valeur à utiliser pour le context paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

Pour URL cible, entrez la valeur à utiliser pour le `target_url` paramètre dans le statut de GitHub validation. Pour plus d'informations, consultez la section [Créer un statut de validation](#) dans le guide du GitHub développeur.

L'état d'une compilation déclenchée par un webhook est toujours communiqué au fournisseur source. Pour que le statut d'une version démarrée depuis la console ou un appel d'API soit signalé au fournisseur source, vous devez sélectionner ce paramètre.

Si les builds de votre projet sont déclenchés par un webhook, vous devez envoyer un nouveau commit au dépôt pour que la modification de ce paramètre prenne effet.

SSL non sécurisé

Sélectionnez Activer le protocole SSL non sécurisé pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub d'entreprise.

Dans Événements de webhook source primaire, sélectionnez Reconstruire chaque fois qu'une modification de code est envoyée à ce référentiel si vous CodeBuild souhaitez générer le code source chaque fois qu'une modification de code est transférée vers ce référentiel. Pour plus d'informations sur les webhooks et les groupes de filtres, consultez [GitHub événements webhook](#).

Environnement

Dans la section Environnement, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Modèle de provisionnement

Pour modifier le modèle de provisionnement, choisissez Modifier le modèle de provisionnement et effectuez l'une des opérations suivantes :

- Pour utiliser des flottes à la demande gérées par AWS CodeBuild, choisissez On-Demand. Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites à la fin de la construction. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.
- Pour utiliser des flottes de capacité réservée gérées par AWS CodeBuild, choisissez Capacité réservée, puis sélectionnez un nom de flotte. Avec les flottes de capacité réservée, vous

configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

Pour plus d'informations, consultez [Travailler avec une capacité réservée dans AWS CodeBuild](#).

Image de l'environnement

Pour modifier l'image de construction, choisissez Remplacer l'image et effectuez l'une des opérations suivantes :

- Pour utiliser une image Docker gérée par AWS CodeBuild, choisissez Managed image (Image gérée), puis effectuez vos sélections pour Système d'exploitation, Exécution, Image et Version d'image. Effectuez votre sélection pour Type d'environnement si cette option est disponible.
- Pour utiliser une autre image Docker, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Si vous choisissez Other registry (Autre registre), pour External registry URL (URL du registre externe), entrez le nom et la balise de l'image Docker dans Docker Hub au format *docker repository/docker image name*. Si vous choisissez Amazon ECR, utilisez le référentiel Amazon ECR et l'image Amazon ECR pour choisir l'image Docker dans votre compte. AWS
- Pour utiliser une image Docker privée, choisissez Image personnalisée. Pour le type d'environnement, choisissez ARM, Linux, Linux GPU ou Windows. Pour Image registry (Registre de l'image), choisissez Other registry (Autre registre) et entrez l'ARN des informations d'identification de votre image Docker privée. Les informations d'identification doivent être créées par Secrets Manager. Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager.

Note

CodeBuild remplace le ENTRYPOINT pour les images Docker personnalisées.

Rôle de service

Effectuez l'une des actions suivantes :

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

- Si vous avez un rôle CodeBuild de service, choisissez Rôle de service existant. Dans Role ARN, choisissez le rôle de service.

Note

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

Configuration supplémentaire

Expiration

Spécifiez une valeur, comprise entre 5 minutes et 8 heures, après quoi la CodeBuild génération s'arrête si elle n'est pas terminée. Si les valeurs de heures et minutes sont laissées vides, la valeur par défaut de 60 minutes est utilisée.

privilégié

Sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos versions bénéficient de privilèges élevés. uniquement si vous prévoyez d'utiliser ce projet de génération pour créer des images Docker. Sinon, toutes les générations associées qui tentent d'interagir avec le démon Docker échouent. Vous devez également démarrer le démon Docker afin que vos générations puissent interagir avec celui-ci. Pour cela, vous pouvez initialiser le démon Docker au cours de la phase `install` de votre spécification de génération en exécutant les commandes de génération ci-après. N'exécutez pas ces commandes si vous avez choisi une image d'environnement de construction fournie CodeBuild par le support Docker.

Note

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

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si vous souhaitez CodeBuild travailler avec votre VPC :

- Pour le VPC, choisissez l'ID du VPC qui utilise CodeBuild
- Pour les sous-réseaux VPC, choisissez les sous-réseaux qui incluent les ressources qui utilisent CodeBuild
- Pour les groupes de sécurité VPC, choisissez les groupes de sécurité CodeBuild utilisés pour autoriser l'accès aux ressources des VPC.

Pour plus d'informations, consultez [Utiliser AWS CodeBuild avec Amazon Virtual Private Cloud](#).

Calcul

Choisissez l'une des options disponibles.

Variables d'environnement

Entrez le nom et la valeur, puis choisissez le type de chaque variable d'environnement à utiliser pour les builds.

Note

CodeBuild définit automatiquement la variable d'environnement pour votre AWS région. Vous devez définir les variables d'environnement suivantes si vous ne les avez pas ajoutées dans votre fichier `buildspec.yml` :

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

Les utilisateurs de console et d'AWS CLI peuvent voir les variables d'environnement. Si la visibilité de vos variables d'environnement ne vous pose pas de problème, définissez les zones Nom et Valeur, puis définissez Type sur Texte brut.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un identifiant de clé d'accès AWS, une clé d'accès AWS secrète ou un mot de passe

en tant que paramètre dans Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager ou.

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, choisissez Parameter dans Type. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value, entrez le nom du paramètre tel qu'il est stocké dans le magasin de paramètres Amazon EC2 Systems Manager. Si l'on prend comme exemple un paramètre nommé `/CodeBuild/dockerLoginPassword`, pour Type, choisissez Parameter (Paramètre). Pour Name (Nom), saisissez LOGIN_PASSWORD. Pour le champ Valeur, saisissez `/CodeBuild/dockerLoginPassword`.

Important

Si vous utilisez Amazon EC2 Systems Manager Parameter Store, nous vous recommandons de stocker les paramètres avec des noms de paramètres commençant `/CodeBuild/` par (par exemple `/CodeBuild/dockerLoginPassword`). Vous pouvez utiliser la CodeBuild console pour créer un paramètre dans Amazon EC2 Systems Manager. Choisissez Create parameter (Créer un paramètre), puis suivez les instructions de la boîte de dialogue. (Dans cette boîte de dialogue, pour la clé KMS, vous pouvez spécifier l'ARN d'une AWS KMS clé dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour la déchiffrer lors de la récupération.) Si vous utilisez la CodeBuild console pour créer un paramètre, la console commence par le nom du paramètre `/CodeBuild/` tel qu'il est enregistré. Pour plus d'informations, consultez la [procédure pas à pas de la console Systems Manager Parameter Store et Systems Manager Parameter Store](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store, le rôle de service du projet de génération doit autoriser `ssm:GetParameters` l'action. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des paramètres stockés dans Amazon EC2 Systems Manager Parameter Store avec des noms de paramètres qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour ce rôle de service pour autoriser l'accès aux noms de

paramètres qui ne commencent pas par `/CodeBuild/`. En effet, ce rôle de service permet uniquement d'accéder aux noms de paramètres qui commencent par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les paramètres sous l'espace de `/CodeBuild/` noms dans le magasin de paramètres Amazon EC2 Systems Manager.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée `MY_VAR` avec la valeur `my_value` et que vous définissez une variable d'environnement nommée `MY_VAR` avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée `PATH` avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée `PATH` avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration `buildspec` a la priorité la plus faible.

Si vous utilisez Secrets Manager, pour Type, choisissez Secrets Manager. Dans Nom, entrez un identifiant CodeBuild à référencer. Pour Value (Valeur), saisissez un `reference-key` à l'aide du modèle `secret-id:json-key:version-stage:version-id`. Pour plus d'informations, consultez [Secrets Manager reference-key in the buildspec file](#).

Important

Si vous utilisez Secrets Manager, nous vous recommandons de stocker les secrets dont le nom commence par `/CodeBuild/` (par exemple, `/CodeBuild/`

`dockerLoginPassword`). Pour plus d'informations, consultez [Présentation de AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager, le rôle de service du projet de génération doit autoriser `secretsmanager:GetSecretValue`. Si vous avez sélectionné Nouveau rôle de service plus tôt, CodeBuild inclut cette action dans le rôle de service par défaut de votre projet de génération. En revanche, si vous avez choisi précédemment Existing service role (Rôle de service existant), vous devez inclure séparément cette action dans votre rôle de service.

Si votre projet de génération fait référence à des secrets stockés dans Secrets Manager avec des noms secrets qui ne commencent pas par `/CodeBuild/`, et que vous avez choisi Nouveau rôle de service, vous devez mettre à jour le rôle de service pour autoriser l'accès aux noms de secret qui ne commencent pas par `/CodeBuild/`. Cela est dû au fait que le rôle de service autorise l'accès uniquement aux noms secrets commençant par `/CodeBuild/`.

Si vous choisissez Nouveau rôle de service, le rôle de service inclut l'autorisation de déchiffrer tous les secrets sous l'espace de `/CodeBuild/` noms dans le Gestionnaire de secrets.

Spécifications de construction

Dans la section Buildspec, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Spécifications de construction

Effectuez l'une des actions suivantes :

- Si votre code source inclut un fichier buildspec, choisissez Utiliser un fichier buildspec. Par défaut, CodeBuild recherche un fichier nommé `buildspec.yml` dans le répertoire racine de code source. Si votre fichier buildspec utilise un nom ou un emplacement différent, entrez son chemin depuis la racine source dans le nom Buildspec (par exemple, `ou.buildspec-two.yml` `configuration/buildspec.yml`). Si le fichier buildspec se trouve dans un compartiment S3, il doit se trouver dans la même AWS région que votre projet de construction. Spécifiez le fichier buildspec à l'aide de son ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

- Si votre code source ne comprend pas de fichier de spécification de génération ou si vous souhaitez exécuter des commandes de génération différentes de celles spécifiées pour la phase `build` dans le fichier `buildspec.yml` au sein du répertoire racine du code source, choisissez Insérer des commandes de génération. Pour Build commands (Commandes de génération), saisissez les commandes que vous souhaitez exécuter lors de la phase `build`. Pour plusieurs commandes, séparez celles-ci avec `&&` (par exemple, `mvn test && mvn package`). Pour exécuter des commandes dans d'autres phases, ou si vous avez une longue liste de commandes pour la `build` phase, ajoutez un `buildspec.yml` fichier dans le répertoire racine du code source, ajoutez les commandes au fichier, puis choisissez Utiliser le fichier `buildspec.yml` dans le répertoire racine du code source.

Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

Configuration par lots

Dans la section Configuration par lots, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration. Pour plus d'informations, consultez [Batch s'intègre AWS CodeBuild](#).

Vous pouvez modifier les propriétés suivantes :

Rôle du service Batch

Fournit le rôle de service pour les compilations par lots.

Sélectionnez l'une des méthodes suivantes :

- Si vous n'avez pas de rôle de service par lots, choisissez Nouveau rôle de service. Dans Rôle de service, entrez le nom du nouveau rôle.
- Si vous avez un rôle de service par lots, choisissez Rôle de service existant. Dans Rôle de service, choisissez le rôle de service.

Les builds par lots introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est requis car CodeBuild il doit être capable d'appeler les `RetryBuild` actions `StartBuild``StopBuild`, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même que celui qu'ils utilisent dans leur build, pour deux raisons :

- L'attribution du rôle `StartBuild` de construction et `RetryBuild` des autorisations permettrait à une seule version de démarrer d'autres versions via le `buildspec`. `StopBuild`

- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds eux-mêmes puissent contourner ces restrictions.

Type (s) de calcul autorisé (s) pour le lot

Sélectionnez les types de calcul autorisés pour le lot. Sélectionnez toutes les réponses qui s'appliquent.

Nombre maximal de builds autorisés par lot

Entrez le nombre maximum de builds autorisés dans le lot. Si un lot dépasse cette limite, il échouera.

Délai d'expiration du Batch

Entrez la durée maximale pendant laquelle la génération par lots doit être terminée.

Combinez des artefacts

Sélectionnez Combiner tous les artefacts du lot en un seul emplacement pour que tous les artefacts du lot soient combinés en un seul emplacement.

Mode de rapport par lots

Sélectionnez le mode de rapport d'état de construction souhaité pour les versions par lots.

Note

Ce champ n'est disponible que lorsque la source du projet est Bitbucket ou GitHub Enterprise GitHub, et l'option Signaler les statuts de construction au fournisseur de source lorsque le début et la fin de vos builds sont sélectionnés sous Source.

Constructions agrégées

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient combinés dans un seul rapport d'état.

Constructions individuelles

Sélectionnez cette option pour que les statuts de toutes les versions du lot soient signalés séparément.

Artefacts

Dans la section Artefacts, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Type

Effectuez l'une des actions suivantes :

- Si vous ne souhaitez pas créer des artefacts de sortie de génération, choisissez Aucun artefact. Vous pouvez le faire si vous exécutez uniquement des tests de compilation ou si vous souhaitez transférer une image Docker vers un référentiel Amazon ECR.
- Pour stocker le résultat du build dans un compartiment S3, choisissez Amazon S3, puis procédez comme suit :
 - Si vous souhaitez utiliser votre nom de projet pour le dossier ou le fichier ZIP de sortie de génération, ne renseignez pas le champ Nom. Sinon, entrez le nom. (Si vous souhaitez produire un fichier ZIP et que vous voulez que celui-ci ait une extension de fichier, veillez à l'inclure après le nom de fichier ZIP.)
 - Sélectionnez Activer la gestion sémantique des versions si vous voulez qu'un nom spécifié dans le fichier buildspec remplace le nom spécifié dans la console. Le nom figurant dans un fichier buildspec est calculé au moment de la génération et utilise le langage de commandes Shell. Par exemple, vous pouvez ajouter une date et une heure au nom de votre artefact afin qu'il soit toujours unique. Les noms d'artefact uniques empêchent les artefacts d'être écrasés. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).
 - Pour Nom du compartiment, choisissez le nom du compartiment de sortie.
 - Si vous avez sélectionné Insérer des commandes de génération précédemment dans cette procédure, pour Fichiers de sortie, saisissez les emplacements des fichiers de la génération que vous souhaitez placer dans le dossier ou le fichier ZIP de sortie de génération. Pour plusieurs emplacements, séparez ceux-ci avec une virgule (par exemple, `appspec.yml, target/my-app.jar`). Pour de plus amples informations, consultez la description de files dans [Syntaxe d'un fichier buildspec](#).
 - Si vous ne souhaitez pas que vos artefacts de génération soient chiffrés, choisissez Remove artifacts encryption (Supprimer le chiffrement des artefacts).

Pour chaque ensemble d'artefacts secondaire que vous souhaitez :

1. Pour Artifact identifier (Identifiant d'artefact), saisissez une valeur de moins de 128 caractères et contenant uniquement des caractères alphanumériques et des traits de soulignement.
2. Choisissez Add artifact (Ajouter un artefact).
3. Suivez les étapes précédentes pour configurer vos artefacts secondaires.
4. Choisissez Save artifact (Enregistrer l'artefact).

Configuration supplémentaire

Clé de chiffrement

Effectuez l'une des actions suivantes :

- Pour utiliser l'Clé gérée par AWSAmazon S3 dans votre compte afin de chiffrer les artefacts de sortie du build, laissez la clé de chiffrement vide. Il s'agit de l'option par défaut.
- Pour utiliser une clé gérée par le client pour chiffrer les artefacts de sortie de génération, dans Clé de chiffrement, entrez l'ARN de la clé gérée par le client. Utilisez le format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Type de cache

Pour Cache type (Type de cache), choisissez l'une des valeurs suivantes :

- Si vous ne souhaitez pas utiliser un cache, choisissez Aucun cache.
- Si vous souhaitez utiliser un cache Amazon S3, choisissez Amazon S3, puis procédez comme suit :
 - Pour Compartiment, choisissez le nom du compartiment S3 dans lequel le cache est stocké.
 - (Facultatif) Pour le préfixe de chemin du cache, entrez un préfixe de chemin Amazon S3. La valeur Cache path prefix (Préfixe du chemin de cache) est semblable à un nom de répertoire. Cela vous permet de stocker le cache sous le même répertoire au sein d'un compartiment.

Important

N'ajoutez pas de barre oblique de fin (/) à la fin du préfixe du chemin.

- Si vous souhaitez utiliser un cache local, choisissez Local, puis sélectionnez une ou plusieurs modes de cache local.

Note

Le mode de cache de couche Docker est disponible pour Linux uniquement. Si vous choisissez ce mode, votre projet doit être exécuté en mode privilégié.

L'utilisation d'un cache permet de gagner beaucoup de temps de génération, car les parties réutilisables de l'environnement de génération sont stockées dans le cache et utilisées d'une génération à l'autre. Pour de plus amples informations sur la spécification d'un cache dans le fichier de spécification de génération, consultez [Syntaxe d'un fichier buildspec](#). Pour plus d'informations sur la mise en cache, consultez [Mise en cache de génération dans AWS CodeBuild](#).

Journaux

Dans la section Journaux, choisissez Modifier. Lorsque vos modifications sont terminées, choisissez Mettre à jour la configuration pour enregistrer la nouvelle configuration.

Vous pouvez modifier les propriétés suivantes :

Choisissez les journaux que vous souhaitez créer. Vous pouvez créer des CloudWatch journaux Amazon Logs, des journaux Amazon S3 ou les deux.

CloudWatch

Si vous souhaitez obtenir CloudWatch les journaux Amazon Logs :

CloudWatch journaux

Sélectionnez Journaux CloudWatch .

Nom du groupe

Entrez le nom de votre groupe de CloudWatch journaux Amazon Logs.

Nom du stream

Entrez le nom de votre flux de journal Amazon CloudWatch Logs.

S3

Si vous souhaitez obtenir les journaux Amazon S3 :

Journaux S3

Sélectionnez Journaux S3.

Compartiment

Choisissez le nom du compartiment S3 pour vos journaux.

Préfixe de chemin

Entrez le préfixe de vos journaux.

Désactiver le chiffrement des journaux S3

Sélectionnez si vous ne voulez pas que vos journaux S3 soient chiffrés.

Modification des paramètres d'un projet de génération (AWS CLI)

Pour de plus amples informations sur l'utilisation de l'AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Pour mettre à jour un CodeBuild projet avec AWS CLI, vous créez un fichier JSON avec les propriétés mises à jour et vous transmettez ce fichier au `update-project` commande. Toutes les propriétés qui ne figurent pas dans le fichier de mise à jour restent inchangées.

Dans le fichier JSON de mise à jour, seul `name` propriété et les propriétés modifiées sont obligatoires. Le `name` La propriété identifie le projet à modifier. Pour toutes les structures modifiées, les paramètres requis pour ces structures doivent également être inclus. Par exemple, pour modifier l'environnement du projet, `environment/type` et `environment/computeType` les propriétés sont obligatoires. Voici un exemple qui met à jour l'image de l'environnement :

```
{
  "name": "<project-name>",
  "environment": {
    "type": "LINUX_CONTAINER",
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:4.0"
  }
}
```

Si vous avez besoin d'obtenir les valeurs de propriété actuelles d'un projet, utilisez le `batch-get-projects` commande pour obtenir les propriétés actuelles du projet que vous modifiez et écrivez le résultat dans un fichier.


```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

Le `project-info.json` Le fichier contient un tableau de projets, il ne peut donc pas être utilisé directement pour mettre à jour un projet. Vous pouvez toutefois copier les propriétés que vous souhaitez modifier à partir du `project-info.json` et collez-les dans votre fichier de mise à jour en tant que référence pour les propriétés que vous souhaitez modifier. Pour plus d'informations, veuillez consulter [Affichage des détails d'un projet de génération \(AWS CLI\)](#).

Modifiez le fichier JSON de mise à jour comme décrit dans [Création d'un projet de génération \(AWS CLI\)](#), et enregistrez vos résultats. Lorsque vous avez terminé de modifier le fichier JSON de mise à jour, exécutez `update-project` commande, en transmettant le fichier JSON de mise à jour.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

En cas de succès, le JSON du projet mis à jour apparaît dans la sortie. Si des paramètres obligatoires sont manquants, un message d'erreur s'affiche dans la sortie pour identifier les paramètres manquants. Par exemple, il s'agit du message d'erreur qui s'affiche si `environment/type` le paramètre est manquant :

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

```
Parameter validation failed:  
Missing required parameter in environment: "type"
```

Modification des paramètres d'un projet de génération (kits SDK AWS)

Pour de plus amples informations sur l'utilisation d'AWS CodeBuild avec le kit SDK AWS, consultez la [Référence des kits SDK et des outils AWS](#).

Suppression d'un projet de génération dans AWS CodeBuild

Vous pouvez utiliser la console CodeBuild, AWS CLI, ou AWS Les kits SDK pour supprimer un projet de génération dans CodeBuild. Si vous supprimez un projet, ses versions ne sont pas supprimées.

⚠ Warning

Vous ne pouvez pas supprimer un projet qui a des générations et une stratégie de ressource. Pour supprimer un projet avec une stratégie de ressources et des générations, supprimez d'abord la stratégie de ressources et ses générations.

Rubriques

- [Suppression d'un projet de génération \(console\)](#)
- [Suppression d'un projet de génération \(AWS CLI\)](#)
- [Suppression d'un projet de génération \(kits SDK AWS\)](#)

Suppression d'un projet de génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Effectuez l'une des actions suivantes :
 - Sélectionnez la case d'option en regard du projet de génération à supprimer, puis choisissez Supprimer.
 - Sélectionnez le lien pour le projet de génération que vous souhaitez supprimer, puis choisissez Supprimer.

i Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour voir plus de projets de génération, choisissez une autre valeur pour Projets per page (Projets par page) ou utilisez les flèches avant et arrière pour consulter les projets.

Suppression d'un projet de génération (AWS CLI)

1. Exécutez la commande `delete-project` :

```
aws codebuild delete-project --name name
```

Remplacez l'espace réservé suivant :

- *nom* : Chaîne obligatoire. Nom du projet de génération à supprimer. Pour obtenir une liste des projets de génération disponibles, exécutez la commande `list-projects`. Pour plus d'informations, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).

2. En cas de réussite, aucune donnée et aucune erreur n'apparaissent dans la sortie.

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Suppression d'un projet de génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Utilisation de projets partagés

Le partage de projets permet aux propriétaires de projets de partager leurs AWS CodeBuild projets avec d'autres AWS comptes ou utilisateurs. Dans ce modèle, le compte propriétaire du projet (propriétaire) partage un projet avec d'autres comptes (consommateurs). Un consommateur ne peut pas modifier ou exécuter un projet.

Table des matières

- [Prérequis pour le partage de projets](#)
- [Conditions préalables à l'accès aux projets partagés avec vous](#)
- [Services connexes](#)
- [Partage d'un projet](#)
- [Annulation du partage d'un projet partagé](#)
- [Identification d'un projet partagé](#)
- [Autorisations de projet partagés](#)

Prérequis pour le partage de projets

Pour partager un projet, votre compte AWS doit le posséder. Vous ne pouvez pas partager un projet qui a été partagé avec vous.

Conditions préalables à l'accès aux projets partagés avec vous

Pour accéder à un projet partagé, le rôle IAM d'un consommateur requiert l'option `BatchGetProjects` autorisation. Vous pouvez attacher la stratégie suivante à leur rôle IAM :

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetProjects"
  ]
}
```

Pour plus d'informations, consultez [Utilisation de politiques basées sur l'identité pour AWS CodeBuild](#).

Services connexes

Le partage de projets s'intègre à AWS Resource Access Manager (AWS RAM), un service qui vous permet de partager vos ressources AWS avec n'importe quel compte AWS ou via AWS Organizations. Avec AWS RAM, vous partagez des ressources en créant un partage de ressources qui spécifie les ressources et les consommateurs avec qui les partager. Les consommateurs peuvent être des comptes individuels AWS, des unités organisationnelles dans AWS Organizations ou l'ensemble d'une organisation dans AWS Organizations.

Pour plus d'informations, veuillez consulter le Guide de l'utilisateur [AWS RAM](#).

Partage d'un projet

Le consommateur peut utiliser à la fois le `aws cli` et `AWS CodeBuild` pour afficher le projet et les générations que vous avez partagés. Le consommateur ne peut pas modifier ou exécuter le projet.

Vous pouvez ajouter un projet à un partage de ressources existant ou en créer un dans la [console AWS RAM](#).

Note


Vous ne pouvez pas supprimer un projet avec des générations qui ont été ajoutées à un partage de ressources.

Pour partager un projet avec des unités organisationnelles ou une organisation entière, vous devez activer le partage avec AWS Organizations. Pour de plus amples informations, veuillez consulter [Activer le partage avec AWS Organizations](#) dans le Guide de l'utilisateur AWS RAM.

Vous pouvez utiliser la console AWS CodeBuild, la console AWS RAM ou l'AWS CLI pour partager un projet que vous possédez.

Pour partager un projet que vous possédez (console CodeBuild)

1. Ouverture d'AWS CodeBuild à <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

 Note

Par défaut, seuls les 10 derniers projets de génération sont affichés. Pour afficher plus de projets de génération, choisissez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Projets par page (Projets par page) ou utilisez les flèches avant et arrière.

3. Choisissez le projet que vous souhaitez partager, puis choisissez Partager. Pour de plus amples informations, veuillez consulter [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM.

Pour partager un projet que vous possédez (console AWS RAM)

Voir [Créer un partage de ressources](#) dans le AWS RAM Guide de l'utilisateur.

Pour partager un projet que vous possédez (commande AWS RAM)

Utilisez la commande [create-resource-share](#).

Pour partager un projet que vous possédez (commande CodeBuild)

Utilisez la commande [put-resource-policy](#) :

1. Créez un fichier nommé `policy.json` et copiez ce qui suit dans celui-ci.

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

```
"Principal":{
  "AWS": "<consumer-aws-account-id-or-user>"
},
"Action":[
  "codebuild:BatchGetProjects",
  "codebuild:BatchGetBuilds",
  "codebuild:ListBuildsForProject"],
"Resource": "<arn-of-project-to-share>"
}]
}
```

2. Mettez à jour `policy.json` avec l'ARN du projet et les identificateurs avec qui le partager. Dans l'exemple suivant, un accès en lecture seule est accordé à l'utilisateur racine pour leAWScompte identifié par 123456789012.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    },
    "Action": [
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource": "arn:aws:codebuild:us-west-2:123456789012:project/my-project"
  }]
}
```

3. Exécutez le [put-resource-policy](#) commande.

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Obtenez leAWS RAMARN de partage de ressources.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Cela renvoie une réponse semblable à ce qui suit :

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
      "lastUpdatedTime": "<last-update-time>"
    }
  ]
}
```

À partir de la réponse, copiez le *<resource-share-arn>* à utiliser à l'étape suivante.

5. Exécutez le AWS RAM [promote-resource-share-created-from-policy](#) commande.

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-share-arn>
```

Annulation du partage d'un projet partagé

Un projet non partagé, y compris ses générations, n'est accessible que par son propriétaire. Si vous désactivez le partage d'un projet, tout compte ou utilisateur AWS avec lequel vous l'avez déjà partagé ne peut pas accéder au projet ni à ses versions.

Pour annuler le partage d'un projet partagé qui vous appartient, vous devez le supprimer du partage de ressources. Vous pouvez utiliser la console AWS CodeBuild, la console AWS RAM ou l'AWS CLI pour ce faire.

Pour annuler le partage d'un projet partagé que vous possédez (console AWS RAM)

Consultez la section [Mise à jour d'un partage de ressources](#) du Guide de l'utilisateur AWS RAM.

Pour annuler le partage d'un projet partagé que vous possédez (AWS CLI)

Utilisez la commande [disassociate-resource-share](#).

Pour éliminer le partage du projet que vous possédez (commande CodeBuild)

Exécutez la commande [delete-resource-policy](#) et spécifiez l'ARN du projet dont vous souhaitez annuler le partage :

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Identification d'un projet partagé

Les propriétaires et les consommateurs peuvent utiliser le AWS CLI pour identifier les projets partagés.

Pour identifier les projets partagés avec votre compte ou utilisateur AWS (AWS CLI)

Utilisez la commande [list-shared-projects](#) pour renvoyer les projets partagés avec vous.

Autorisations de projet partagés

Autorisations accordées aux propriétaires

Un propriétaire de projet peut modifier le projet et l'utiliser pour exécuter des générations.

Autorisations accordées aux consommateurs

Un consommateur de projet peut afficher un projet et ses versions, mais ne peut pas modifier un projet ou l'utiliser pour exécuter des générations.

Balissage de projets dans AWS CodeBuild

Une balise est un attribut personnalisé que vous attribuez ou qu'AWS attribue à une ressource AWS. Chaque balise AWS se compose de deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment`, `Project` ou `Secret`). Les clés de balises sont sensibles à la casse.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333`, `Production` ou le nom d'une équipe). Omettre la valeur de balise équivaut à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Ensemble, ces éléments sont connus sous le nom de paires clé-valeur. Pour plus d'informations sur le nombre de balises que vous pouvez avoir sur un projet et les restrictions sur les clés et les valeurs de balises, veuillez consulter [Balises](#).

Les balises vous aident à identifier et organiser vos ressources AWS. De nombreux services AWS prennent en charge le balissage. Vous pouvez donc attribuer la même balise à des ressources à partir de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer

la même balise à un projet CodeBuild que celle vous affectez à un compartiment S3. Pour de plus amples informations sur l'utilisation des balises sur l'utilisation des balises sur l'utilisation [des balises](#), consultez [Balisage](#).

Dans CodeBuild, les ressources principales sont le projet et le groupe de rapports. Vous pouvez utiliser la console CodeBuild, l'AWS CLI, les API CodeBuild ou les kits SDK AWS pour ajouter, gérer et supprimer des balises d'un projet. Outre l'identification, l'organisation et le suivi de votre projet avec des balises, vous pouvez utiliser des balises dans les politiques IAM afin de contrôler qui peut consulter et interagir avec votre projet. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Rubriques

- [Ajouter une balise à un projet](#)
- [Afficher les balises d'un projet](#)
- [Modifier les balises d'un projet](#)
- [Supprimer une balise d'un projet](#)

Ajouter une balise à un projet

L'ajout de balises à un projet peut vous aider à identifier et organiser vos ressources AWS et à gérer leur accès. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à un projet. N'oubliez pas qu'il y a des limites sur le nombre de balises que vous pouvez avoir dans un projet. Il existe des restrictions sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour plus d'informations, veuillez consulter [Balises](#). Une fois que vous avez défini des balises, vous pouvez créer des politiques IAM pour gérer l'accès au projet en fonction de ces balises. Vous pouvez utiliser la console CodeBuild ou l'AWS CLI pour ajouter des balises à un référentiel.

Important

Avant d'ajouter une balise à un projet, assurez-vous de consulter les politiques IAM susceptibles d'utiliser des balises pour contrôler l'accès à des ressources telles que des projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Pour plus d'informations sur l'ajout de balises à un projet lorsque vous le créez, consultez [Ajouter une balise à un projet \(console\)](#).

Rubriques

- [Ajouter une balise à un projet \(console\)](#)
- [Ajouter une balise à un projet \(AWS CLI\)](#)

Ajouter une balise à un projet (console)

Vous pouvez utiliser la console CodeBuild pour ajouter une ou plusieurs balises à un projet CodeBuild.

1. Ouvrez la CodeBuild console à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Dans Build projects (Créer des projets), choisissez le nom du projet auquel vous souhaitez ajouter des balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Si aucune balise n'a été ajoutée au projet, choisissez Ajouter une balise. Sinon, choisissez Edit (modifier), puis choisissez Add tag (Ajouter une balise).
5. Dans Key (Clé), entrez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).
6. (Facultatif) Pour ajouter une autre balise, choisissez à nouveau Add tag (Ajouter une balise).
7. Lorsque vous avez fini d'ajouter des balises, choisissez Submit (Envoyer).

Ajouter une balise à un projet (AWS CLI)

Pour ajouter une balise à un pipeline lors de sa création, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#). Dans `create-project.json`, ajoutez vos balises.

Dans ces étapes, nous supposons que vous avez déjà installé une version récente de l'AWS CLI ou que vous avez procédé à une mise à jour vers la version actuelle. Pour plus d'informations, consultez [Installing the AWS Command Line Interface](#) (Installation de).

Si elle aboutit, cette commande ne renvoie rien.

Afficher les balises d'un projet

Les balises peuvent vous aider à identifier et organiser vos ressources AWS et à gérer leur accès. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc sur les

[bonnes pratiques de balisage](#). Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Afficher les balises d'un projet (console)

Vous pouvez utiliser la console CodeBuild pour afficher les balises associées à un projet CodeBuild.

1. Ouvrez la CodeBuild console à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Dans Build projects (Créer des projets), choisissez le nom du projet dans lequel vous souhaitez afficher les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).

Afficher les balises d'un projet (AWS CLI)

Pour afficher les balises d'un projet de génération, exécutez la commande suivante. Utilisez le nom de votre projet pour le paramètre `--names`.

```
aws codebuild batch-get-projects --names your-project-name
```

Si elle réussit, cette commande renvoie des informations au format JSON sur votre projet de construction qui incluent quelque chose comme ce qui suit :

```
{
  "tags": {
    "Status": "Secret",
    "Team": "JanesProject"
  }
}
```

Si le projet n'a pas de balises, la section `tags` est vide :

```
"tags": []
```

Modifier les balises d'un projet

Vous pouvez modifier la valeur d'une balise associée à un projet. Vous pouvez également modifier le nom de la clé, ce qui équivaut à supprimer la balise et à ajoutant une carte différente avec le nouveau

nom et la même valeur que l'autre clé. N'oubliez pas qu'il y a des limites sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour plus d'informations, veuillez consulter [Balises](#).

Important

La modification des balises d'un projet peut avoir un impact sur l'accès au projet. Avant de modifier le nom (clé) ou la valeur d'une balise pour un projet, assurez-vous de consulter toutes les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès à des ressources telles que des projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Modifier une balise d'un projet (console)

Vous pouvez utiliser la console CodeBuild pour modifier les balises associées à un projet CodeBuild.

1. Ouvrez la CodeBuild console à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Dans Build projects (Créer des projets), choisissez le nom du projet dont vous souhaitez modifier les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Choisissez Edit (Modifier).
5. Effectuez l'une des actions suivantes :
 - Pour modifier la balise, entrez un nouveau nom dans Key (Clé). La modification du nom de la balise est l'équivalent de la suppression d'une balise et de l'ajout d'une nouvelle balise avec le nouveau nom de clé.
 - Pour modifier la valeur d'une balise, saisissez une nouvelle valeur. Si vous souhaitez modifier la valeur en valeur nulle, supprimez la valeur actuelle et laissez le champ vide.
6. Lorsque vous avez terminé de modifier des balises, choisissez Submit (Soumettre).

Modifier les balises d'un projet (AWS CLI)

Pour ajouter, modifier ou supprimer des balises d'un projet de génération, veuillez consulter [Modification des paramètres d'un projet de génération \(AWS CLI\)](#). Mettez à jour la section tags des données au format JSON que vous utilisez pour mettre à jour le projet.

Supprimer une balise d'un projet

Vous pouvez supprimer une ou plusieurs balises associées à un projet. La suppression d'une balise ne supprime pas la balise d'autres ressources AWS qui sont associées à cette balise.

Important

La suppression des balises d'un projet peut avoir un impact sur l'accès au projet. Avant de supprimer une balise d'un projet, assurez-vous de consulter toutes les politiques IAM susceptibles d'utiliser la clé ou la valeur d'une balise pour contrôler l'accès à des ressources telles que des projets de création. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Supprimer une balise d'un projet (console)

Vous pouvez utiliser la console CodeBuild pour supprimer l'association entre une balise et un projet CodeBuild.

1. Ouvrez la CodeBuild console à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Dans Build projects (Créer des projets), choisissez le nom du projet dans lequel vous souhaitez supprimer les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Choisissez Build project tags (Créer des balises de projet).
4. Choisissez Edit (Modifier).
5. Trouvez la balise que vous voulez supprimer, puis choisissez Remove tag (Supprimer une balise).
6. Lorsque vous avez terminé de supprimer les balises, choisissez Submit (Envoyer).

Supprimer une balise d'un projet (AWS CLI)

Pour supprimer une ou plusieurs balises d'un projet de génération, reportez-vous à la section [Modification des paramètres d'un projet de génération \(AWS CLI\)](#). Mettez à jour la tags section dans les données formatées JSON avec une liste mise à jour de balises qui ne contient pas celles que vous souhaitez supprimer. Si vous souhaitez supprimer toutes les balises, mettez à jour la section tags pour :

```
"tags: []"
```

Note

Si vous supprimez un projet de génération CodeBuild, toutes les associations de balises sont supprimées du projet de génération supprimé. Vous n'avez pas besoin de supprimer les balises avant de supprimer un projet de génération.

Batch s'intègre AWS CodeBuild

Vous pouvez l'utiliser AWS CodeBuild pour exécuter des versions simultanées et coordonnées d'un projet avec des versions par lots.

Rubriques

- [Rôle de sécurité](#)
- [Types de build par lots](#)
- [Mode de rapport par lots](#)
- [En savoir plus](#)

Rôle de sécurité

Les builds par lots introduisent un nouveau rôle de sécurité dans la configuration par lots. Ce nouveau rôle est requis car CodeBuild il doit être capable d'appeler les `RetryBuild` actions `StartBuild``StopBuild`, et en votre nom pour exécuter des builds dans le cadre d'un lot. Les clients doivent utiliser un nouveau rôle, et non le même que celui qu'ils utilisent dans leur build, pour deux raisons :

- L'attribution du rôle `StartBuild` de construction et `RetryBuild` des autorisations permettrait à une seule version de démarrer d'autres versions via le `buildspec`. `StopBuild`
- CodeBuild les versions par lots fournissent des restrictions qui limitent le nombre de versions et les types de calcul qui peuvent être utilisés pour les versions du lot. Si le rôle de build dispose de ces autorisations, il est possible que les builds eux-mêmes puissent contourner ces restrictions.

Types de build par lots

CodeBuild prend en charge les types de génération par lots suivants :

Types de build par lots

- [Construire un graphe](#)
- [Construire une liste](#)
- [Construire une matrice](#)

Construire un graphe

Un graphe de génération définit un ensemble de tâches qui dépendent d'autres tâches du lot.

L'exemple suivant définit un graphe de construction qui crée une chaîne de dépendances.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
```

Dans cet exemple :

- build1 s'exécute en premier car il n'a aucune dépendance.
- build2 dépend de build1, donc build2 s'exécute une fois build1 terminé.

- build3 dépend de build2, donc build3 s'exécute une fois build2 terminé.

Pour plus d'informations sur la syntaxe buildspec du graphe de construction, consultez [batch/build-graph](#)

Construire une liste

Une liste de build définit un certain nombre de tâches exécutées en parallèle.

L'exemple suivant définit une liste de build. Les build2 builds build1 et seront exécutés en parallèle.

```
batch:
  fast-fail: false
  build-list:
    - identifiant: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifiant: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
```

Pour plus d'informations sur la syntaxe buildspec de la liste de compilation, consultez [batch/build-list](#)

Construire une matrice

Une matrice de génération définit les tâches avec différentes configurations qui s'exécutent en parallèle. CodeBuild crée une version distincte pour chaque combinaison de configuration possible.

L'exemple suivant montre une matrice de construction avec deux fichiers buildspec et trois valeurs pour une variable d'environnement.

```
batch:
  build-matrix:
    static:
      ignore-failure: false
```



```
dynamic:
  buildspec:
    - matrix1.yml
    - matrix2.yml
  env:
    variables:
      MY_VAR:
        - VALUE1
        - VALUE2
        - VALUE3
```

Dans cet exemple, CodeBuild crée six versions :

- matrix1.yml avec \$MY_VAR=VALUE1
- matrix1.yml avec \$MY_VAR=VALUE2
- matrix1.yml avec \$MY_VAR=VALUE3
- matrix2.yml avec \$MY_VAR=VALUE1
- matrix2.yml avec \$MY_VAR=VALUE2
- matrix2.yml avec \$MY_VAR=VALUE3

Chaque build comportera les paramètres suivants :

- ignore-failureréglé sur false
- env/typeréglé sur LINUX_CONTAINER
- env/imageréglé sur aws/codebuild/amazonlinux2-x86_64-standard:4.0
- env/privileged-moderéglé sur true

Ces builds s'exécutent en parallèle.

Pour plus d'informations sur la syntaxe buildspec de la matrice de construction, consultez [batch/build-matrix](#)

Mode de rapport par lots

Si le fournisseur source de votre projet est Bitbucket ou GitHub Enterprise GitHub, et que votre projet est configuré pour signaler les statuts de construction au fournisseur source, vous pouvez sélectionner la manière dont vous souhaitez que les statuts de génération par lots soient envoyés au fournisseur source. Vous pouvez choisir d'envoyer les statuts sous forme de rapport d'état

agrégé unique pour le lot, ou de faire en sorte que le statut de chaque build du lot soit signalé individuellement.

Pour plus d'informations, consultez les rubriques suivantes :

- [Configuration par lots \(création\)](#)
- [Configuration par lots \(mise à jour\)](#)

En savoir plus

Pour plus d'informations, consultez les rubriques suivantes :

- [Référence de la spécification de construction par Batch](#)
- [Configuration par lots](#)
- [Exécution d'une génération par lots \(AWS CLI\)](#)
- [Arrêt d'une génération dans AWS CodeBuild](#)

GitHubAction Runner dans AWS CodeBuild

Vous pouvez utiliser un Action Runner pour courir GitHub Actions au sein de CodeBuild. Cela peut être fait en ajoutant des étapes à n'importe quelle phase dans votre fichier buildspec.

CodeBuild buildspecs supporte une liste de séquences GitHub Étapes d'action exécutées dans une phase distincte de CodeBuild commandes. Ces GitHub Les actions s'intègrent à CodeBuild les fonctionnalités existantes, notamment la mise en cache des dépendances, les builds par lots, l'accès à AWS Secrets Manager, et bien plus encore.

Rubriques

- [Comment démarrer avec le GitHubAction Runner ?](#)
- [Lequel GitHub Les actions que je peux utiliser ?](#)
- [Puis-je utiliser des fournisseurs de sources autres que GitHub lors de l'utilisation GitHub Actions dans mon CodeBuild projet ?](#)
- [Pourquoi dois-je me connecter à GitHub en tant que fournisseur de sources afin d'utiliser GitHub Des actions ?](#)
- [Combien coûte l'utilisation du GitHubAction Runner ?](#)
- [Quelles régions soutiennent le GitHubAction Runner ?](#)

- [Les meilleures pratiques pour GitHub Actions](#)
- [Limites du GitHub Action Runner dans CodeBuild](#)
- [GitHub Exemples d'actions avec AWS CodeBuild](#)

Comment démarrer avec le GitHub Action Runner ?

Les étapes de haut niveau pour ajouter un GitHub Les actions sont les suivantes :

1. Si ce n'est pas déjà fait, connectez votre projet à GitHub.

Pour ce faire, vous pouvez effectuer l'une des opérations suivantes :

- Vous pouvez ajouter GitHub en tant que fournisseur de source dans la console. Pour plus d'informations, veuillez consulter [Connect GitHub avec un jeton d'accès \(console\)](#).
- Vous pouvez importer votre GitHub informations d'identification via le [CodeBuild API](#). Pour plus d'informations, veuillez consulter [Connectez-vous à GitHub l'aide d'un jeton d'accès \(CLI\)](#).

Note

Cela ne doit être fait que si vous n'êtes pas connecté à GitHub dans un autre projet.

2. Dans les spécifications de construction de votre projet, vous pouvez ajouter `steps`, dont chacun fait référence à un `GitHub Action`. Cela peut être modifié dans le `CodeBuild console` ou dans votre référentiel source. Chaque phase de construction prend en charge une liste de commandes ou une liste d'étapes, mais les deux ne peuvent pas être utilisées dans la même phase. Pour plus d'informations, veuillez consulter [GitHub Référence de spécification de construction d'Action Runner](#).

Lequel GitHub Les actions que je peux utiliser ?

Vous pouvez utiliser n'importe quelle action disponible dans le [GitHub Marketplace](#) cela n'entre pas en conflit avec ceux-ci [limitations](#).

Puis-je utiliser des fournisseurs de sources autres que GitHub lors de l'utilisation GitHub Actions dans mon CodeBuild projet ?

Oui, mais en se connectant à GitHub est toujours nécessaire pour s'authentifier auprès de GitHub et accéder à GitHub Actions. Pour plus d'informations, veuillez consulter [GitHub et jeton d'accès au serveur GitHub Enterprise](#).

Pourquoi dois-je me connecter à GitHub en tant que fournisseur de sources afin d'utiliser GitHub Actions ?

Afin d'utiliser GitHub Actions, la source doit être téléchargée sur un ordinateur de compilation. Le débit des téléchargements anonymes sera limité. Par conséquent, en vous connectant à GitHub, cela peut contribuer à garantir un accès cohérent.

Combien coûte l'utilisation du GitHub Action Runner ?

Courir GitHub Actions sont soutenues sans frais supplémentaires.

Quelles régions soutiennent le GitHub Action Runner ?

GitHub Actions sont soutenues dans tous CodeBuild régions. Pour plus d'informations sur Régions AWS où CodeBuild est disponible, voir [AWS Services par région](#).

Les meilleures pratiques pour GitHub Actions

GitHub Actions sont open source, créées et maintenues par la communauté. Nous suivons le [modèle de responsabilité partagée](#) et considérez GitHub Code source des actions en tant que données clients dont vous êtes responsable. GitHub Actions peuvent être autorisées à accéder aux secrets, aux jetons du référentiel, au code source et aux liens de compte. Assurez-vous d'avoir confiance en la fiabilité et en la sécurité de GitHub Actions que vous comptez exécuter.

Des conseils plus spécifiques et des meilleures pratiques de sécurité pour GitHub Actions :

- [Renforcement de la sécurité](#)
- [Empêcher ses propres demandes](#)
- [Entrée non fiable](#)
- [Comment faire confiance à vos éléments de base](#)

Limites du GitHub Action Runner dans CodeBuild

- GitHub Actions qui s'appuient en interne sur [le githubcontexte](#) ou cette référence GitHub-les ressources spécifiques, telles que les pull requests et les problèmes, ne sont pas prises en charge dans CodeBuild. Par exemple, les actions suivantes ne fonctionneront pas dans CodeBuild:
 - GitHub Actions visant à ajouter, modifier ou mettre à jour GitHub ressources, telles que les actions qui mettent à jour les pull requests ou créent des problèmes dans GitHub.

Note

Le plus officiel GitHub Actions répertoriées dans <https://github.com/actions> compter sur [githubcontexte](#). Utilisez plutôt les actions disponibles dans le [GitHub Marketplace](#).

- GitHub Des actions qui sont [Actions du conteneur Docker](#) fonctionnera, mais votre projet de construction doit avoir [mode privilégié](#) activé et exécuté par l'utilisateur Docker par défaut (root).
 - Les actions doivent être exécutées en tant qu'utilisateur root. Pour plus d'informations, consultez le [UTILISATEUR](#) sujet dans [Support de Dockerfile pour GitHub Actions](#).
- GitHub Les actions ne sont pas prises en charge dans CodeBuild projets configurés pour fonctionner sous Windows.
- GitHub Tâches d'action (groupes d'étapes) et GitHub Les propriétés des tâches d'action ne sont pas prises en charge.
- GitHub Les actions ne sont pas prises en charge dans CodeBuild projets configurés pour être déclenchés par un webhook pour un dépôt Git public. Pour plus d'informations, voir [git-credential-helper](#).
- Les builds VPC sans accès public à Internet ne peuvent pas s'exécuter GitHub Des actions.
- Chaque phase de construction prend en charge une liste de commandes ou une liste d'étapes, mais les deux ne peuvent pas être utilisées dans la même phase. Par exemple, dans l'exemple suivant, des étapes sont utilisées dans la phase de pré-construction pour répertorier GitHub Actions, tandis que les commandes sont utilisées dans la phase de construction pour répertorier CodeBuild commandes.

```
version: 0.2
phases:
  pre-build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v4
```

```
env:
  VALIDATE_ALL_CODEBASE: 'true'
  DEFAULT_BRANCH: main
build:
  commands:
    - echo "Building..."
    - npm run build
```

GitHub Exemples d'actions avec AWS CodeBuild

Ces groupes d'échantillons peuvent être utilisés pour expérimenter avec GitHub Actions dans CodeBuild.

Rubriques

- [Super-Linter GitHub Exemple d'action](#)
- [Exemple de graphique de création par lots](#)
- [Amazon CodeGuru Exemple de réviseur](#)
- [Exemple AWS Secrets Manager](#)
- [Exemple de variable d'environnement](#)
- [Exemple de variable d'environnement exporté](#)

Super-Linter GitHub Exemple d'action

Cet exemple montre comment ajouter le [Super-Linter](#) GitHub Action en faveur d'un CodeBuild projet. L'action Super-Linter inspecte le code, détecte les zones dans lesquelles le code contient des erreurs, des problèmes de mise en forme et des constructions suspectes, puis affiche les résultats vers CodeBuild console.

Vous pouvez ajouter le Super-Linter GitHub Action pour votre CodeBuild projet en mettant à jour la section de phase de votre fichier buildspec.

```
version: 0.2
phases:
  build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v5
        env:
```

```
VALIDATE_ALL_CODEBASE: true
```

Les journaux Super-Linter ressembleront à ce qui suit :

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.  
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.  
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body' found.  
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but found  
4.  
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not  
found.
```

Exemple de graphique de création par lots

L'exemple suivant définit un graphe de construction qui crée une chaîne de dépendances et exécute des commandes à l'aide de `steps`. Dans cet exemple, `build1` s'exécute en premier car il n'a aucune dépendance. Depuis `build2` dépend de `build1`, donc `build2` s'exécute une fois que `build1` est terminé. Pour plus d'informations, consultez [Construire un graphe](#).

```
version: 0.2  
batch:  
  fast-fail: false  
  build-graph:  
    - identifier: build1  
      env:  
        variables:  
          BUILD_ID: build1  
      ignore-failure: false  
    - identifier: build2  
      env:  
        variables:  
          BUILD_ID: build2  
      depend-on:  
        - build1  
  
phases:  
  build:  
    steps:  
      - run: echo $BUILD_ID
```

AmazonCodeGuruExemple de réviseur

AmazonCodeGuruLe réviseur détecte les problèmes dans votre code Java et Python et recommande comment y remédier. L'exemple suivant utiliseCodeGuruRéviseur chargé de fournir des révisions complètes du code d'analyse du référentiel. Ces révisions de code scannent tout le code d'une branche spécifiée. Pour plus d'informations, voir[Créez des révisions de code avecGitHubActions](#)dans leAmazonCodeGuruGuide de l'utilisateur du réviseur.

```
version: 0.2
phases:
  build:
    steps:
      - name: Amazon CodeGuru Reviewer Scanner
        if: ${{ always() }}
        uses: aws-actions/codeguru-reviewer@v1.1
        with:
          s3_bucket: codeguru-reviewer-user

artifacts:
  files:
    - codeguru-results.sarif.json
```

Note

Votre compartiment Amazon S3 doit commencer parcodeguru-reviewer-préfixe.

Les journaux ressembleront à ce qui suit :

```
INFO CodeReview created with arn=arn:aws:codeguru-reviewer:region:account-
id:association:id:code-review:RepositoryAnalysis-job for job=job
INFO SARIF persisted to /github/workspace/codeguru-results.sarif.json
INFO Amazon CodeGuru Reviewer job execution completed
```

Après l'AmazonieCodeGuruLe travail du réviseur est terminé, un rapport sarif est généré sous la formeCodeBuildartefact. Pour plus d'informations, voir[Analyse complète du référentiel](#)dans leAmazonCodeGuruGuide de l'utilisateur du réviseur.

Exemple AWS Secrets Manager

AWS Secrets Manager vous aide à gérer, récupérer et faire pivoter les informations d'identification des bases de données, les informations d'identification des applications, les jetons OAuth, les clés d'API et d'autres secrets tout au long de leur cycle de vie. L'exemple suivant définit un secret à l'aide de Secrets Manager et exécute des commandes à l'aide des `steps`. Pour plus d'informations, voir [Qu'est-ce que AWS Secrets Manager?](#) dans le [AWS Secrets Manager Guide de l'utilisateur](#).

```
version: 0.2
env:
  secrets-manager:
    SECRET_VALUE: "arn:aws:secretsmanager:us-east-1:xxxx:secret:/secret-
13IJg9:my_super_secret_key"
phases:
  build:
    steps:
      - run: echo $SECRET_VALUE
```

Les journaux ressembleront à ce qui suit :

```
echo $SECRET_VALUE
env:
  SECRET_VALUE: ***
***
```

Exemple de variable d'environnement

L'exemple suivant définit les variables d'environnement sous `env` séquence. `UNS3_BUCKET` la variable est définie dans le `buildspec` et attribuée `<bucket-name>` comme sa valeur. Cette variable est référencée dans le `if` conditionnel comme une variable d'environnement normale en utilisant le signe dollar (\$) pour accéder au `GitHubContext` de l'environnement d'action. Pour plus d'informations, voir [env séquence](#).

```
version: 0.2
env:
  variables:
    S3_BUCKET: "<bucket-name>"
phases:
  build:
    steps:
      - if: ${{ env.S3_BUCKET == '<bucket-name>' }}
```

```
run: echo "S3 bucket is $S3_BUCKET"
```

Les journaux ressembleront à ce qui suit :

```
echo "S3 bucket is $S3_BUCKET"
env:
  S3_BUCKET: my-s3-bucket
S3 bucket is my-s3-bucket
```

Exemple de variable d'environnement exporté

Les variables d'environnement exportées sont utilisées conjointement avec `CodePipeline` pour exporter les variables d'environnement de la phase de construction actuelle vers les étapes suivantes du pipeline. L'exemple suivant définit une variable d'environnement exportée sous `envs` séquence nommée `MON_VARIABLE` et écrit au `GITHUB_ENV` fichier d'environnement.

```
version: 0.2
env:
  exported-variables:
    - MY_VARIABLE
phases:
  build:
    steps:
      - run: echo "MY_VARIABLE=my-value" >> $GITHUB_ENV
```

Pour plus d'informations, consultez [ExportedEnvironmentVariable](#) dans la Référence d'API AWS CodeBuild.

Projets de build public dans AWS CodeBuild

AWS CodeBuild vous permet de mettre à la disposition du grand public les résultats de la génération, les journaux et les artefacts de vos projets de construction. Cela permet aux contributeurs de vos référentiels sources d'afficher les résultats et de télécharger les artefacts d'une build, sans qu'ils aient accès à un AWS.

Lorsque vous rendez les builds de votre projet accessibles au public, tous les résultats de construction, les journaux et les artefacts d'un projet, y compris les builds exécutés lorsque le projet était privé, sont mis à la disposition du public. De même, lorsque vous rendez un projet de construction public privé, les résultats de construction de ce projet ne sont plus accessibles au public.

Pour plus d'informations sur la modification de la visibilité publique des résultats de génération de votre projet, consultez [Permettre l'accès public aux builds](#).

CodeBuild fournit une URL pour les builds publics de votre projet spécifique à votre projet. Pour obtenir l'URL publique de votre projet de génération, exécutez la procédure suivante :

1. Ouverture d'AWS CodeBuild console à <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Sélectionnez le lien pour le projet de génération pour lequel vous souhaitez obtenir l'URL publique.
4. L'URL publique est affichée dans le URL du projet public dans le champ Configuration Section. Vous pouvez choisir le lien pour ouvrir l'URL ou copier l'URL à l'aide du bouton Copier.

Warning

Les éléments suivants doivent être gardés à l'esprit lorsque vous publiez les résultats de génération de votre projet :

- Tous les résultats de construction, les journaux et les artefacts d'un projet, y compris les builds exécutés lorsque le projet était privé, sont accessibles au public.
- Tous les journaux de construction et les artefacts sont accessibles au public. Les variables d'environnement, le code source et d'autres informations sensibles peuvent avoir été générés dans les journaux de génération et les artefacts. Vous devez être prudent quant aux informations générées dans les journaux de génération. Voici quelques bonnes pratiques :
 - Ne stockez pas les valeurs sensibles, en particulier AWS les ID de clé d'accès et les clés d'accès secrètes dans des variables d'environnement. Nous vous recommandons d'utiliser un magasin de paramètres Amazon EC2 Systems Manager ou AWS Secrets Manager pour stocker des valeurs sensibles.
 - Suivre [Bonnes pratiques d'utilisation des webhooks](#) pour limiter les entités pouvant déclencher une génération et ne pas stocker la spécification de construction dans le projet lui-même, afin de garantir que vos webhooks sont aussi sécurisés que possible.
 - Un utilisateur malveillant peut utiliser des versions publiques pour distribuer des artefacts malveillants. Nous recommandons aux administrateurs de projet de passer en revue toutes

les demandes d'extraction afin de vérifier que la pull request est une modification légitime. Nous vous recommandons également de valider tous les artefacts avec leurs sommes de contrôle pour vous assurer que les artefacts corrects sont en cours de téléchargement.

Gestion des générations dans AWS CodeBuild

Une génération représente un ensemble d'actions exécutées par AWS CodeBuild pour créer des artefacts de sortie (par exemple, un fichier JAR) basé sur un ensemble d'artefacts d'entrée (par exemple, une collection de fichiers de classe Java).

Les règles suivantes s'appliquent lorsque vous exécutez plusieurs builds :

- Les builds s'exécutent simultanément, lorsque possible. Le nombre maximal de versions en cours d'exécution simultanée peut varier. Pour plus d'informations, consultez [Quotas pour AWS CodeBuild](#).
- Si le projet de génération comporte une limite de génération simultanée, les builds renvoient une erreur si le nombre de builds en cours d'exécution atteint la limite de génération simultanée pour le projet. Pour de plus amples informations, veuillez consulter [Activer la limite de construction simultanée](#).
- Si le projet de construction n'a pas de limite de génération simultanée, les builds sont mis en file d'attente si le nombre de builds en cours d'exécution atteint la limite de génération simultanée pour la plate-forme et le type de calcul. Le nombre maximal de builds dans une file d'attente est à cinq fois la limite des builds simultanés. Pour plus d'informations, consultez [Quotas pour AWS CodeBuild](#).

Un build dans une file d'attente qui ne démarre pas après le nombre de minutes spécifié dans sa valeur d'expiration est supprimé de la file d'attente. La valeur de délai d'expiration par défaut est de huit heures. Vous pouvez remplacer le délai de mise en file d'attente de génération par une valeur comprise entre cinq minutes et huit heures lorsque vous exécutez votre build. Pour plus d'informations, consultez [Exécution d'une génération dans AWS CodeBuild](#).

Il n'est pas possible de prédire l'ordre dans lequel les builds mis en attente démarrent.

Note

Vous pouvez accéder à l'historique d'une génération pour une durée d'un an.

Vous pouvez effectuer ces tâches lorsque vous gérez des générations :

Rubriques

- [Exécution d'une génération dans AWS CodeBuild](#)
- [Affichage des détails d'une génération dans AWS CodeBuild](#)
- [Affichage d'une liste d'ID de génération dans AWS CodeBuild](#)
- [Affichage d'une liste d'ID de génération pour un projet de génération dans AWS CodeBuild](#)
- [Arrêt d'une génération dans AWS CodeBuild](#)
- [Arrêt d'une génération dans AWS CodeBuild](#)
- [Retente une génération dans AWS CodeBuild](#)
- [Afficher une version en cours d'exécution dans le Gestionnaire de session](#)
- [Suppression de générations dans AWS CodeBuild](#)

Exécution d'une génération dans AWS CodeBuild

Vous pouvez utiliser le plugin `AWS CodeBuild console` AWS CLI, ou `AWS Kits SDK` pour exécuter une génération dans CodeBuild.

Rubriques

- [Exécution d'une génération \(console\)](#)
- [Exécution d'une génération \(AWS CLI\)](#)
- [Exécution d'une génération par lots \(AWS CLI\)](#)
- [Lancement automatique de l'exécution de générations \(AWS CLI\)](#)
- [Arrêt automatique de l'exécution de générations \(AWS CLI\)](#)
- [Exécution d'une génération \(kits SDK AWS\)](#)

Exécution d'une génération (console)

Pour utiliser `AWS CodePipeline` pour exécuter une génération avec CodeBuild, ignorez ces étapes et suivez les instructions de [À utiliser CodePipeline avec CodeBuild](#).

1. Ouverture d'`AWS CodeBuild console` <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de génération, choisissez le projet de génération.

4. Vous pouvez exécuter la génération avec les paramètres de projet de génération par défaut ou remplacer les paramètres de génération pour cette génération uniquement.
 - a. Pour que exécuter la génération avec les paramètres par défaut du projet de génération, choisissez **Démarrer la génération**. La génération commence immédiatement.
 - b. Pour que remplacer les paramètres par défaut du projet de génération, choisissez **Démarrer la génération avec des remplacements**. Dans **Démarrer la génération**, vous pouvez remplacer les éléments suivants :
 - Build configuration (Configuration de la génération)
 - Source
 - Remplacements de variable d'environnement

Si vous devez sélectionner des remplacements plus avancés, choisissez **Remplacements avancés de génération**. Sur cette page, vous pouvez remplacer les éléments suivants :

- Build configuration (Configuration de la génération)
- Source
- Environnement
- BuildSpec
- Artefacts
- Journaux

Lorsque vous avez effectué vos sélections de remplacement, choisissez **Démarrer la génération**.

Pour de plus amples informations sur cette génération, consultez [Affichage des détails d'une génération \(console\)](#).

Exécution d'une génération (AWS CLI)

Note

Pour utiliser CodePipeline afin d'exécuter une génération avec AWS CodeBuild, ignorez ces étapes et suivez les instructions de [Création d'un pipeline utilisant CodeBuild \(AWS CLI\)](#).

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec CodeBuild, consultez [Référence des commandes en ligne](#).

1. Exécutez la commande `start-build` de l'une des manières suivantes :

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

Utilisez cette commande si vous souhaitez exécuter une génération utilisant la dernière version de l'artefact d'entrée de génération et les paramètres existants du projet de génération.

```
aws codebuild start-build --generate-cli-skeleton
```

Utilisez cette commande si vous souhaitez exécuter une génération avec une version précédente de l'artefact d'entrée de génération, ou si vous souhaitez remplacer les paramètres pour les artefacts de sortie, les variables d'environnement, le fichier `buildspec` ou le délai d'expiration de génération par défaut.

2. Si vous exécutez la commande `start-build` avec l'option `--project-name`, remplacez `<project-name>` avec le nom du projet de construction, puis passez à l'étape 6 de cette procédure. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération](#).
3. Si vous exécutez la commande `start-build` avec l'option `--idempotency-token`, un identifiant ou un jeton unique distinguant majuscules et minuscules, est incluse dans la commande. Le jeton est valide pendant 5 minutes après la demande. Si vous répétez la demande `start-build` avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance de paramètre.
4. Si vous exécutez la commande `start-build` avec l'option `--generate-cli-skeleton`, des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, `start-build.json`) dans un emplacement de l'ordinateur local ou de l'instance locale où l'interface AWS CLI est installée. Modifiez les données copiées pour qu'elles correspondent au format suivant, et enregistrez vos résultats :

```
{
  "projectName": "projectName",
  "sourceVersion": "sourceVersion",
  "artifactsOverride": {
    "type": "type",
```

```
"location": "location",
"path": "path",
"namespaceType": "namespaceType",
"name": "artifactsOverride-name",
"packaging": "packaging"
},
"buildspecOverride": "buildspecOverride",
"cacheOverride": {
  "location": "cacheOverride-location",
  "type": "cacheOverride-type"
},
"certificateOverride": "certificateOverride",
"computeTypeOverride": "computeTypeOverride",
"environmentTypeOverride": "environmentTypeOverride",
"environmentVariablesOverride": {
  "name": "environmentVariablesOverride-name",
  "value": "environmentVariablesValue",
  "type": "environmentVariablesOverride-type"
},
"gitCloneDepthOverride": "gitCloneDepthOverride",
"imageOverride": "imageOverride",
"idempotencyToken": "idempotencyToken",
"insecureSslOverride": "insecureSslOverride",
"privilegedModeOverride": "privilegedModeOverride",
"queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
"reportBuildStatusOverride": "reportBuildStatusOverride",
"timeoutInMinutesOverride": "timeoutInMinutesOverride",
"sourceAuthOverride": "sourceAuthOverride",
"sourceLocationOverride": "sourceLocationOverride",
"serviceRoleOverride": "serviceRoleOverride",
"sourceTypeOverride": "sourceTypeOverride"
}
```

Remplacez les espaces réservés suivants :

- *projectName* : chaîne obligatoire. Nom du projet de génération à utiliser pour la génération.
- *sourceVersion* : chaîne facultative. Une version du code source à générer, comme suit :
 - Pour Amazon S3, l'ID de version qui correspond à la version du fichier ZIP d'entrée que vous souhaitez créer. Si *sourceVersion* n'est pas spécifié, la dernière version est utilisée.
 - Pour CodeCommit, l'ID de validation qui correspond à la version du code source à générer. Si *sourceVersion* n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est

utilisé. (Vous ne pouvez pas spécifier un nom de balise *sourceVersion* mais vous pouvez spécifier l'ID de validation de la balise.)

- Pour GitHub, l'ID de validation, l'ID de pull request, le nom de branche ou le nom de balise correspondant à la version du code source que vous souhaitez créer. Si un ID de demande d'extraction est spécifié, il doit utiliser le format `pr/pull-request-ID` (par exemple, `pr/25`). Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. Si *sourceVersion* n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Pour Bitbucket, l'ID de validation, le nom de branche ou le nom de balise qui correspond à la version du code source à générer. Si un nom de branche est spécifié, l'ID de validation HEAD de la branche est utilisé. Si *sourceVersion* n'est pas spécifié, l'ID de validation HEAD de la branche par défaut est utilisé.
- Les espaces réservés suivants sont pour `artifactsOverride`.
 - *type* : facultatif. Type d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *location* : facultatif. Emplacement d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *path* : facultatif. Chemin d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *namespaceType* : facultatif. Type de chemin d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *name* : facultatif. Nom d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
 - *packaging* : facultatif. Type de packaging d'artefact de sortie de génération qui remplace pour cette génération celui défini dans le projet de génération.
- *buildspecOverride* : facultatif. Déclaration de spécification de génération (buildspec) qui remplace pour cette génération celle définie dans le projet de génération. Si cette valeur est définie, il peut s'agir d'une définition de spécification de génération en ligne, du chemin vers un fichier buildspec alternatif, relatif à la valeur de la variable d'environnement `CODEBUILD_SRC_DIR` intégrée, ou du chemin vers un compartiment S3. Le compartiment S3 doit être dans la même région AWS que le projet de génération. Spécifiez le fichier buildspec à l'aide de son nom ARN (par exemple, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Si cette valeur n'est pas fournie ou qu'elle est définie sur une chaîne vide,

le code source doit contenir un fichier `buildspec.yml` dans son répertoire racine. Pour plus d'informations, veuillez consulter [Nom de fichier buildspec et emplacement de stockage](#).

- Les espaces réservés suivants sont pour `cacheOverride`.
 - *cacheOverride-location* : facultatif. Emplacement d'un objet `ProjectCache` pour cette génération qui remplace l'objet `ProjectCache` spécifié dans le projet de génération. `cacheOverride` est facultatif et accepte un objet `ProjectCache`. `location` est obligatoire dans un objet `ProjectCache`.
 - *cacheOverride-type* : facultatif. Type d'un objet `ProjectCache` pour cette génération qui remplace l'objet `ProjectCache` spécifié dans le projet de génération. `cacheOverride` est facultatif et accepte un objet `ProjectCache`. `type` est obligatoire dans un objet `ProjectCache`.
- *certificateOverride* : facultatif. Nom d'un certificat pour cette génération qui remplace celui spécifié dans le projet de génération.
- *environmentTypeOverride*: Facultatif. Type de conteneur pour cette génération qui remplace celui spécifié dans le projet de génération. La chaîne valide actuelle est `LINUX_CONTAINER`.
- Les espaces réservés suivants sont pour `environmentVariablesOverride`.
 - *environmentVariablesOverride-nom*: Facultatif. Nom d'une variable d'environnement dans le projet de génération dont vous voulez substituer la valeur pour cette génération.
 - *environmentVariablesOverride-type*: Facultatif. Type de variable d'environnement dans le projet de génération dont vous voulez substituer la valeur pour cette génération.
 - *environmentVariablesValue*: Facultatif. Valeur de la variable d'environnement définie dans le projet de génération que vous voulez substituer pour cette génération.
- *gitCloneDepthReplacer*: Facultatif. Valeur de `Git clone depth` dans le projet de génération dont vous voulez remplacer la valeur pour cette génération. Si votre type de source est Amazon S3, cette valeur n'est pas prise en charge.
- *imageOverride* : facultatif. Nom d'une image pour cette génération qui remplace celle spécifiée dans le projet de génération.
- *idempotencyToken* : facultatif. Chaîne qui tient lieu de jeton pour spécifier que la demande de génération est idempotente. Vous pouvez choisir n'importe quelle chaîne de 64 caractères maximum. Le jeton est valide pendant 5 minutes après la demande de démarrage de la construction. Si vous répétez la demande de démarrage de la construction avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance des paramètres.

- ***insecureSslOverride***: booléen facultatif qui indique s'il faut remplacer le paramètre TLS non sécurisé spécifié dans le projet de génération. Le paramètre TLS non sécurisé détermine s'il convient d'ignorer les avertissements TLS lors de la connexion au code source du projet. Cette dérogation ne s'applique que si la source de la version est `GitHubServeur d'entreprise`.
- ***privilegedModeOverride***: booléen facultatif. Si celle-ci est définie sur `true`, la génération remplace le mode privilégié dans le projet de génération.
- ***queuedTimeoutInMinutesOverride***: entier facultatif qui indique le nombre de minutes pendant lesquelles une version est autorisée à être mise en file d'attente avant son expiration. Sa valeur minimale est de cinq minutes et sa valeur maximale est 480 minutes (huit heures).
- ***reportBuildStatusReplacer***: booléen facultatif qui indique s'il faut envoyer à votre fournisseur source l'état du début et de la fin d'une compilation. Si vous l'avez défini avec un fournisseur de source autre que `GitHub`, `GitHubEnterprise Server`, ou `Bitbucket`, `uninvalidInputException` est jeté.
- ***sourceAuthOverride***: chaîne facultative. Type d'autorisation pour cette génération qui remplace celle définie dans le projet de génération. Cette dérogation s'applique uniquement si la source du projet de build est `Bitbucket` ou `GitHub`.
- ***sourceLocationOverride***: chaîne facultative. Emplacement qui remplace l'emplacement de la source pour cette génération par celui défini dans le projet de génération.
- ***serviceRoleOverride***: chaîne facultative. Nom d'un rôle de service pour cette génération qui remplace celui spécifié dans le projet de génération.
- ***sourceTypeOverride***: chaîne facultative. Type d'entrée source pour cette génération qui remplace l'entrée source définie dans le projet de génération. Les chaînes valides sont `NO_SOURCE`, `CODECOMMIT`, `CODEPIPELINE`, `GITHUB`, `S3`, `BITBUCKET` et `GITHUB_ENTERPRISE`.
- ***timeoutInMinutesReplacer***: Numéro facultatif. Nombre de minutes de délai d'expiration de génération qui remplace pour cette de génération celui défini dans le projet de génération.

Nous vous recommandons de stocker une variable d'environnement avec une valeur sensible, telle qu'un `AWSID` de clé d'accès, et `AWS` clé d'accès secrète ou mot de passe en tant que paramètre dans le magasin de paramètres Amazon EC2 Systems Manager. CodeBuild peut utiliser un paramètre stocké dans le magasin de paramètres Amazon EC2 Systems Manager uniquement si le nom de ce paramètre commence par `/CodeBuild/` (par exemple, `/CodeBuild/dockerLoginPassword`). Vous pouvez utiliser le `CodeBuild` console pour créer un paramètre dans Amazon EC2 Systems Manager. Sélectionnez `Créer un paramètre`, puis

suivez les instructions. (Dans cette boîte de dialogue, pourclé KMS, vous pouvez éventuellement spécifier l'ARN d'unAWS KMSentrez dans votre compte. Amazon EC2 Systems Manager utilise cette clé pour chiffrer la valeur du paramètre pendant le stockage et pour le déchiffrer lors de la récupération.) Si vous utilisez la console CodeBuild pour créer un paramètre, cette dernière fait commencer le nom du paramètre par /CodeBuild/ lors de son stockage. Toutefois, si vous utilisez la console Amazon EC2 Systems Manager Parameter Store pour créer un paramètre, vous devez commencer le nom du paramètre par/CodeBuild/, et vous devez définirTypepourChaîne sécurisée. Pour plus d'informations, voir[AWS Systems Managermagasin de paramètres](#)et[Procédure pas à pas : création et test d'un paramètre de chaîne \(console\)](#)dans leGuide de l'utilisateur d'Amazon EC2 Systems Manager.

Si votre projet de construction fait référence à des paramètres stockés dans le magasin de paramètres Amazon EC2 Systems Manager, le rôle de service du projet de construction doit autoriser`ssm:GetParameters`action. Si vous avez choisi précédemment Créer un rôle de service dans votre compte, CodeBuild inclut automatiquement cette action dans le rôle de service par défaut de votre projet de génération. Toutefois, si vous avez choisi précédemment Choisir un rôle de service existant dans votre compte, vous devez inclure séparément cette action dans votre rôle de service.

Les variables d'environnement que vous définissez remplacent les variables d'environnement existantes. Par exemple, si l'image Docker contient déjà une variable d'environnement nommée MY_VAR avec la valeur `my_value` et que vous définissez une variable d'environnement nommée MY_VAR avec la valeur `other_value`, la valeur `my_value` est remplacée par `other_value`. De même, si l'image Docker contient déjà une variable d'environnement nommée PATH avec la valeur `/usr/local/sbin:/usr/local/bin` et que vous définissez une variable d'environnement nommée PATH avec la valeur `$PATH:/usr/share/ant/bin`, la valeur `/usr/local/sbin:/usr/local/bin` est remplacée par la valeur littérale `$PATH:/usr/share/ant/bin`.

Ne définissez pas de variables d'environnement avec un nom commençant par `CODEBUILD_`. Ce préfixe est réservé à une utilisation interne .

Si une variable d'environnement avec le même nom est définie dans plusieurs emplacements, la valeur de la variable d'environnement est déterminée comme suit :

- La valeur de l'appel d'opération de démarrage de génération a une priorité plus élevée.
- La valeur de la définition de projet de génération vient ensuite dans l'ordre des priorités.
- La valeur figurant dans la déclaration du fichier `buildspec` a la priorité la plus faible.

Pour de plus amples informations sur les valeurs valides pour ces espaces réservés, consultez [Création d'un projet de génération \(AWS CLI\)](#). Pour obtenir une liste des derniers paramètres pour un projet de génération, consultez [Affichage des détails d'un projet de génération](#).

5. Placez-vous dans le répertoire contenant le fichier que vous venez d'enregistrer, puis exécutez à nouveau la commande `start-build`.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. En cas de réussite, des données similaires à ce qui est décrit dans la procédure [Pour exécuter la génération](#) apparaissent en sortie.

Pour utiliser les informations détaillées sur cette génération, notez la valeur `id` de la sortie, puis consultez [Affichage des détails d'une génération \(AWS CLI\)](#).

Exécution d'une génération par lots (AWS CLI)

1. Exécutez la commande `start-build-batch` de l'une des manières suivantes :

```
aws codebuild start-build-batch --project-name <project-name>
```

Utilisez cette commande si vous souhaitez exécuter une génération utilisant la dernière version de l'artefact d'entrée de génération et les paramètres existants du projet de génération.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Utilisez cette commande si vous souhaitez exécuter une génération avec une version précédente de l'artefact d'entrée de génération, ou si vous souhaitez remplacer les paramètres pour les artefacts de sortie, les variables d'environnement, le fichier `buildspec` ou le délai d'expiration de génération par défaut.

2. Si vous exécutez `start-build-batch` avec la commande `--project-name`, remplacer `<project-name>` avec le nom du projet de génération, puis passez directement à l'étape 6 de cette procédure. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération](#).
3. Si vous exécutez `start-build-batch` avec la commande `--idempotency-token`, un identifiant sensible à la casse unique, ou jeton, est inclus avec `start-build-batch` de la demande.

Le jeton est valide pendant 5 minutes après la demande . Si vous répétez le `start-build-batch` avec le même jeton, mais que vous modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance de paramètre.

4. Si vous exécutez le `start-build-batch` avec la commande `--generate-cli-skeleton`, les données au format JSON sont affichées dans le `<json-file>` dans le fichier. Ce fichier est similaire à celui produit par le `start-build`, avec l'ajout de l'objet suivant. Pour en savoir plus sur les objets communs, consultez [Exécution d'une génération \(AWS CLI\)](#).

Modifiez ce fichier pour ajouter des remplacements de génération, et enregistrez les résultats.

```
"buildBatchConfigOverride": {
  "combineArtifacts": combineArtifacts,
  "restrictions": {
    "computeTypesAllowed": [
      allowedComputeTypes
    ],
    "maximumBuildsAllowed": maximumBuildsAllowed
  },
  "serviceRole": "batchServiceRole",
  "timeoutInMins": batchTimeout
}
```

Le `buildBatchConfigOverride` l'objet est un [ProjectBuildBatchConfig](#) qui contient la configuration de build par lots remplace cette version.

CombineArtifacts

Booléen qui spécifie si les artefacts de génération pour la génération par lots doivent être combinés dans un seul emplacement d'artefact.

Types de calcul autorisés

Tableau de chaînes qui spécifient les types de calcul autorisés pour la construction par lots. Voir [Types de calcul d'environnement de génération](#) pour ces valeurs.

MaximumBuilds Allowed

Spécifie le nombre maximal de builds autorisés.

Rôle de service par lots

Spécifie l'ARN du rôle de service pour le projet de génération par lots.

Délai d'expiration du lot

Spécifie la durée maximale, en minutes, pendant laquelle la génération du lot doit être terminée.

5. Placez-vous dans le répertoire contenant le fichier que vous venez d'enregistrer, puis exécutez à nouveau la commande `start-build-batch`.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. En cas de succès, la représentation JSON d'un [BuildBatch](#)s'affiche dans la sortie de la console. Consultez [Syntaxe de la réponse StartBuildBatch](#) pour un exemple de ces données.

Lancement automatique de l'exécution de générations (AWS CLI)

Si votre code source est stocké dans un référentiel GitHub ou GitHub Enterprise Server, vous pouvez utiliser les webhooks GitHub pour qu'AWS CodeBuild génère à nouveau votre code source chaque fois qu'un changement de code est transmis en mode push au référentiel.

Exécutez la commande `create-webhook` comme suit :

```
aws codebuild create-webhook --project-name <project-name>
```

<project-name> représente le nom du projet de génération contenant le code source à régénérer.

Pour GitHub, des informations similaires à ce qui suit s'affichent dans la sortie :

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

<url> représente l'URL du webhook GitHub.

Pour GitHub Enterprise Server, des informations similaires à ce qui suit s'affichent dans la sortie :

```
{
  "webhook": {
    "secret": "YRV4JYAGFsekJiirp5ytx86oZpyhUdySNSDTLNUxoXX1c7aZ6XYDf37-ZFY02rs4JSE70mLW3w-gh-ryoVB80SS5C1aAtBtuPkHwYuncCCmdogCVCfniQ7ukYX2_xM--n1Dma5EngIg_Bi_N465yi33zyTUNPoQ1xCpLO-BwghcVa91AurwR77-uY7i-_XCJFahwMx1f4ubOgBB5mMT2A16apqJqQJoKSb61XVKyZy1Giuy4nliAXFv9WmN76CaCsndb3fVIE78fpygfo41xYxSQ6vpo6LRTKtPzbyeTHbVXGda1PjvnkBlmKmjDo0RTgI1m2oYr17dwziQ1rrvoCoNgy1S00_7LKfA-nNXFc_f1SiFy0AqeMB43-d00cdkzybHncE81QTRwEUCFfmX-AJCwmLXV0kg0G67T92Sjbpz0fRlkh5pwIF193_bB_j0HDinK6i0iPpf2dIDAIZgGMagqZeWb-axDeTABopoU8J6gFI1yKo5aq9q151zC1PERUsMgJfTjr_a-z-L_ky1r-4hSSxasSJNuJ43_XOBRWqT51xqvH-A69bv07KbVT_Kc6wxkSHyYCEMoa_Pfa7ZQgyfY6B00ogMNj31yFbjthORNL1cDo6-3J-McDLoyrRtSEOV9QnxvsG5zu1N5-z20rkJtg_M0fNwocfUutFXb7vrGTduH1R1dzXLRusHuXOVVuDUUm9vhwMr-hUkeGo_1kDKyk4E2QFvZXpjYw0vFfV-dwxFRR_miFzxW1wyfmt2iFtLkp_YZj_4WeFackGefr-i1NaYvsZpzXj78Ae1adVoLf48AmDdN2pWswJjatU9zt942gLisFFmKakcvJuySyxXHaxxbhUyC8NHYiESUWPfcfnqrMsr8op3P4AUCHIpiZCYyuiwI_cac-pIUB00Xaur_lu_fyFghg0Jc7cfTnA36rv5X5DnFDM8P3HNBeLjaF9QZ6AijegPEWTHIKJON3AUDwpkz_hwTXyUoAU8MdZfPTXbBoT6N5Z5THBHsYxR",
    "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?t=eyJlbnNyeXB0ZWREYXRhIjo1UmFqMmJERGRQbGhwLzNTN1d3R0VGRjZzOTNwLzLzVG1Nz1pIR1E0RUsxdzhGeWhnVFFqWTR0WEFwT2dJRnNmRHc3S3Rnc0xYMEhNcXFTagk1cE1nSy9zPSIsIm12UGFyYW1ldGVyU3B1YyI6IndSQ1Qrc2VpQjBCZzhPeYyilCJtYXRlcm1hbFNldFNlcm1hbCI6Mm90X3D&v=1"
  }
}
```

1. Copiez la clé secrète et l'URL de la charge utile à partir de la sortie. Vous en aurez besoin lorsque vous ajouterez un webhook dans GitHub Enterprise Server.
2. Dans GitHub Enterprise Server, choisissez le référentiel dans lequel votre projet CodeBuild est stocké. Choisissez Settings (Paramètres), Hooks & services, puis Add webhook (Ajouter un webhook).
3. Entrez l'URL et la clé secrète de la charge utile, acceptez les valeurs par défaut pour les autres champs, puis choisissez Add webhook.

Arrêt automatique de l'exécution de générations (AWS CLI)

Si votre code source est stocké dans un référentiel GitHub ou GitHub Enterprise Server, vous pouvez configurer des webhooks GitHub afin qu'AWS CodeBuild génère à nouveau votre code source chaque fois qu'un changement de code est transmis en mode push au référentiel. Pour plus d'informations, consultez [Lancement automatique de l'exécution de générations \(AWS CLI\)](#).

Si vous avez activé ce comportement, vous pouvez le désactiver en exécutant la commande `delete-webhook` comme suit :

```
aws codebuild delete-webhook --project-name <project-name>
```

- où `<project-name>` représente le nom du projet de génération contenant le code source à régénérer.

Si cette commande aboutit, aucune information ni aucune erreur n'apparaît dans la sortie.

Note

Cette commande supprime le webhook uniquement dans votre projet CodeBuild. Vous devez également supprimer le webhook de votre référentiel GitHub ou GitHub Enterprise Server.

Exécution d'une génération (kits SDK AWS)

Pour utiliser CodePipeline pour exécuter une build avec AWS CodeBuild, ignorez ces étapes et suivez les instructions de [Utilisation d'AWS CodePipeline avec AWS CodeBuild pour tester le code et exécuter des générations](#) à la place.

Pour plus d'informations sur l'utilisation de CodeBuild avec les kits SDK AWS, veuillez consulter la [Référence des kits SDK et des outils AWS](#).

Affichage des détails d'une génération dans AWS CodeBuild

Vous pouvez utiliser le plugin AWS CodeBuild console AWS CLI, ou les kits SDK AWS pour afficher des détails sur les générations gérées par CodeBuild.

Rubriques

- [Affichage des détails d'une génération \(console\)](#)
- [Affichage des détails d'une génération \(AWS CLI\)](#)
- [Affichage des détails d'une génération \(kits SDK AWS\)](#)
- [Transitions des phases de génération](#)

Affichage des détails d'une génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, dans la colonne Exécution de génération, choisissez le lien de la génération.
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien du nom du projet de génération. Puis, dans la liste des générations, dans la colonne Exécution de génération, choisissez le lien de la génération.

Note

Par défaut, seuls les 10 derniers projets de génération ou générations s'affichent. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projects per page (Projets par page) ou utilisez les flèches avant et arrière.

Affichage des détails d'une génération (AWS CLI)

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Exécutez la commande batch-get-builds :

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

Remplacez l'espace réservé suivant :

- *identifiants* : Chaîne obligatoire. Un ou plusieurs ID de génération sur lesquels afficher les détails. Pour spécifier plusieurs ID de génération, séparez-les par un espace. Vous pouvez spécifier jusqu'à 100 ID de génération. Pour obtenir une liste d'ID de générations, consultez les rubriques suivantes :
 - [Affichage d'une liste d'ID de génération \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#)

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

Si la commande aboutit, des données similaires à celles décrites dans [Pour afficher les informations de génération résumées](#) apparaissent dans la sortie.

Affichage des détails d'une génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Transitions des phases de génération

Les générations dans AWS CodeBuild sont exécutées dans des phases :



⚠ Important

La phase `UPLOAD_ARTIFACTS` est toujours tentée, même si la phase `BUILD` échoue.

Affichage d'une liste d'ID de génération dans AWS CodeBuild

Vous pouvez utiliser le plugin `AWS CodeBuild console` AWS CLI, ou `AWS kits SDK` pour afficher une liste d'ID de génération pour les générations gérées par CodeBuild.

Rubriques

- [Affichage d'une liste d'ID de génération \(console\)](#)
- [Affichage d'une liste d'ID de génération \(AWS CLI\)](#)
- [Affichage d'une liste d'ID de génération par lots \(AWS CLI\)](#)
- [Affichage d'une liste d'ID de génération \(kits SDK AWS\)](#)

Affichage d'une liste d'ID de génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Dans le volet de navigation, choisissez Historique de génération.

Note

Par défaut, seules les 10 dernières générations s'affichent. Pour afficher plus de générations, sélectionnez l'icône en forme d'engrenage, puis sélectionnez une autre valeur pour Générations par page ou utilisez les flèches avant et arrière.

Affichage d'une liste d'ID de génération (AWS CLI)

Pour plus d'informations sur l'utilisation du, consultez [AWS CLI avec CodeBuild](#), consultez le [Référence des commandes en ligne](#).

- Exécutez la commande `list-builds` :

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- sort-order* : chaîne facultative utilisée pour indiquer comment afficher la liste des ID de génération. Les valeurs valides sont ASCENDING et DESCENDING.
- jeton suivant* : chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild list-builds --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
```

```

"codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
"codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
... The full list of build IDs has been omitted for brevity ...
"codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}

```

Si vous exécutez à nouveau cette commande :

```

aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full
token has been omitted for brevity...MzY20A==

```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```

{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}

```

Affichage d'une liste d'ID de génération par lots (AWS CLI)

Pour plus d'informations sur l'utilisation du, consultez [AWS CLI avec CodeBuild](#), consultez le [Référence des commandes en ligne](#).

- Exécutez la commande `list-build-batches` :

```

aws codebuild list-build-batches --sort-order sort-order --next-token next-token

```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *sort-order* : chaîne facultative utilisée pour indiquer comment afficher la liste des ID de génération par lots. Les valeurs valides sont ASCENDING et DESCENDING.
- *jeton suivant* : chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste,

exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Un résultat similaire à ce qui suit peut s'afficher dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Affichage d'une liste d'ID de génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation de CodeBuild avec l', consultez [AWS Kits SDK](#), consultez le [Référence des kits SDK et des outils AWS](#).

Affichage d'une liste d'ID de génération pour un projet de génération dans AWS CodeBuild

Vous pouvez utiliser le plugin [AWS CodeBuild console AWS CLI](#), ou [AWS Kits SDK](#) pour afficher une liste d'ID de génération pour un projet de génération dans CodeBuild.

Rubriques

- [Affichage d'une liste d'ID de génération pour un projet de génération \(console\)](#)
- [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#)
- [Affichage d'une liste d'ID de génération par lots pour un projet de génération \(AWS CLI\)](#)
- [Affichage d'une liste d'ID de génération pour un projet de génération \(kits SDK AWS\)](#)

Affichage d'une liste d'ID de génération pour un projet de génération (console)

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le projet de génération.

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projets par page (Projets par page) ou utilisez les flèches avant et arrière.

Affichage d'une liste d'ID de génération pour un projet de génération (AWS CLI)

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Exécutez la commande `list-builds-for-project`, comme suit :

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *project-name* : chaîne obligatoire utilisée pour indiquer le nom du projet de génération pour lequel répertorier les ID de génération. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- *pour l'ordre de tri* : chaîne facultative utilisée pour indiquer comment afficher la liste des ID de génération. Les valeurs valides sont ASCENDING et DESCENDING.
- *jeton suivant* : chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur renvoyé, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez cette commande comme suit :

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :


```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --
sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY2OA==
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Affichage d'une liste d'ID de génération par lots pour un projet de génération (AWS CLI)

Pour plus d'informations sur l'utilisation de l'interface AWS CLI avec AWS CodeBuild, consultez [Référence des commandes en ligne](#).

Exécutez la commande `list-build-batches-for-project`, comme suit :

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-
order sort-order --next-token next-token
```

Dans la commande précédente, remplacez les espaces réservés suivants :

- *project-name* : chaîne obligatoire utilisée pour indiquer le nom du projet de génération pour lequel répertorier les ID de génération. Pour obtenir une liste de projets de génération, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- *pour l'ordre de tri* : chaîne facultative utilisée pour indiquer comment afficher la liste des ID de génération. Les valeurs valides sont ASCENDING et DESCENDING.
- *jeton suivant* : chaîne facultative. Lors d'une exécution précédente, si la liste contenait plus de 100 éléments, seuls les 100 premiers éléments étaient renvoyés, ainsi qu'une chaîne unique appelée jeton suivant. Pour obtenir l'ensemble d'éléments suivant dans la liste, exécutez à nouveau cette commande, en ajoutant le jeton suivant à l'appel. Pour obtenir tous les éléments de la liste, continuez d'exécuter cette commande avec chaque jeton suivant ultérieur renvoyé, jusqu'à ce qu'aucun jeton suivant ne soit renvoyé.

Par exemple, si vous exécutez cette commande comme suit :

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --
sort-order ASCENDING
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Si vous exécutez à nouveau cette commande :

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project
--sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY2OA==
```

Un résultat similaire à ce qui suit peut apparaître dans la sortie :

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Affichage d'une liste d'ID de génération pour un projet de génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez

[Référence des kits SDK et des outils AWS.](#)

Arrêt d'une génération dans AWS CodeBuild

Vous pouvez utiliser la console AWS CodeBuild, l'AWS CLI, ou des kits SDK AWS pour arrêter une génération dans AWS CodeBuild.

Rubriques

- [Arrêt d'une génération \(console\)](#)
- [Arrêt d'une génération \(AWS CLI\)](#)
- [Arrêt d'une génération \(kits SDK AWS\)](#)

Arrêt d'une génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si la page **nom-projet-génération:ID-génération** s'affiche, choisissez Arrêter la génération.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projets par page (Projets par page) ou utilisez les flèches avant et arrière.

Si AWS CodeBuild ne parvient pas arrêter une génération (par exemple, si le processus de génération est déjà terminé), le bouton Arrêter est désactivé voire absent.

Arrêt d'une génération (AWS CLI)

- Exécutez la commande stop-build :

```
aws codebuild stop-build --id id
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- *identifiant* : Chaîne obligatoire. ID de la génération à arrêter. Pour obtenir une liste d'ID de générations, consultez les rubriques suivantes :
 - [Affichage d'une liste d'ID de génération \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#)

Si AWS CodeBuild arrête avec succès la génération, la valeur `buildStatus` de l'objet `build` de la sortie est STOPPED.

Si CodeBuild ne parvient pas arrêter la génération (par exemple, si la génération est déjà terminée), le `buildStatus` valeur de l'objet `build` dans la sortie correspond à l'état final de la génération (par exemple, SUCCEEDED).

Arrêt d'une génération (kits SDK AWS)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Arrêt d'une génération dans AWS CodeBuild

Vous pouvez utiliser le plugin `aws codebuild console` AWS CLI, ou `aws kits` SDK pour arrêter une génération par lots dans AWS CodeBuild.

Rubriques

- [Arrêt d'une génération par lots \(console\)](#)
- [Arrêt d'une génération par lots \(AWS CLI\)](#)
- [Arrêt d'une génération par lots \(AWS Kits SDK\)](#)

Arrêt d'une génération par lots (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si la page ***nom-projet-génération:ID-génération*** s'affiche, choisissez Arrêter la génération.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Stop build (Arrêter la génération).

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projets par page (Projets par page) ou utilisez les flèches avant et arrière.

Si AWS CodeBuild ne peut pas arrêter une génération par lots (par exemple, si le processus de génération est déjà terminé), l'arrêt de génération est désactivé.

Arrêt d'une génération par lots (AWS CLI)

- Exécutez la commande [stop-build-batch](#) :

```
aws codebuild stop-build-batch --id <batch-build-id>
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- ***<batch-build-id>*** : Chaîne obligatoire. Identifiant de la génération par lots à arrêter. Pour obtenir une liste d'identifiants de génération par lots, consultez les rubriques suivantes :
 - [Affichage d'une liste d'ID de génération par lots \(AWS CLI\)](#)

- [Affichage d'une liste d'ID de génération par lots pour un projet de génération \(AWS CLI\)](#)

Si AWS CodeBuild arrête avec succès la génération par lots, le `buildBatchStatus` valeur dans le `buildBatch` objet dans la sortie est `STOPPED`.

Si CodeBuild ne peut pas arrêter la génération par lots (par exemple, si la génération par lots est déjà terminée), le `buildBatchStatus` valeur dans le `buildBatch` objet dans la sortie correspond à l'état final de la génération (par exemple, `SUCCEEDED`).

Arrêt d'une génération par lots (AWS Kits SDK)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Retente une génération dans AWS CodeBuild

Vous pouvez utiliser le plugin AWS CodeBuild console AWS CLI, ou AWS SDK pour réessayer une version unique ou une génération par lots AWS CodeBuild.

Rubriques

- [Retente une génération \(console\)](#)
- [Retente une génération \(AWS CLI\)](#)
- [Retente une génération \(AWS Kits SDK\)](#)

Retente une génération (console)

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Effectuez l'une des actions suivantes :
 - Si l'icône **nom-projet de construction : ID de builds** s'affiche, choisissez Retente la génération.
 - Dans le volet de navigation, choisissez Historique de génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez Retente la génération.
 - Dans le volet de navigation, choisissez Projets de génération. Dans la liste des projets de génération, dans la colonne Nom, choisissez le lien correspondant au nom du projet de

génération. Dans la liste des générations, cochez la case correspondant à la génération, puis choisissez la génération.

Note

Par défaut, seuls les 100 derniers projets de génération ou générations sont affichés. Pour afficher plus de générations ou de projets de génération, sélectionnez l'icône en forme d'engrenage, puis choisissez une autre valeur pour Générations par page ou Projets par page (Projets par page) ou utilisez les flèches avant et arrière.

Retente une génération (AWS CLI)

- Exécutez la commande `retry-build` :

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- <build-id>** : Chaîne obligatoire. ID de la génération ou de la génération par lots à nouvelle tentative. Pour obtenir une liste d'ID de générations, consultez les rubriques suivantes :
 - [Affichage d'une liste d'ID de génération \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération par lots \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération par lots pour un projet de génération \(AWS CLI\)](#)
- idempotency-token** - Facultatif. Si vous exécutez `retry-build` avec l'option, un identifiant sensible à la casse unique, ou jeton, est inclus avec `retry-build` de la demande. Le jeton est valide pendant 5 minutes après la demande . Si vous répétez `retry-build` avec le même jeton, mais modifiez un paramètre, CodeBuild renvoie une erreur de non-concordance de paramètre.

Retente une génération (AWS Kits SDK)

Pour plus d'informations sur l'utilisation d'AWS CodeBuild avec les kits SDK AWS, consultez [Référence des kits SDK et des outils AWS](#).

Afficher une version en cours d'exécution dans le Gestionnaire de session

Dans AWS CodeBuild, vous pouvez suspendre une compilation en cours, puis utiliser le gestionnaire de AWS Systems Manager session pour vous connecter au conteneur de génération et afficher l'état du conteneur.

Note

Cette fonctionnalité n'est pas disponible dans les environnements Windows.

Rubriques

- [Prérequis](#)
- [Suspendre le build](#)
- [Démarrez la construction](#)
- [Connectez-vous au conteneur de construction](#)
- [Reprendre le build](#)

Prérequis

Pour autoriser l'utilisation du gestionnaire de session avec la session de génération, vous devez activer la connexion à la session pour la génération. Il y a deux prérequis :

- CodeBuild L'agent SSM est déjà installé et l'agent SSM est activé sur les images sélectionnées standard de Linux. ContainerMode

Si vous utilisez une image personnalisée pour votre build, procédez comme suit :

1. Installez SSM Agent. Pour plus d'informations, consultez la section [Installation manuelle de l'agent SSM sur les instances EC2 pour Linux](#) dans le guide de l' AWS Systems Manager utilisateur. La version de l'agent SSM doit être 3.0.1295.0 ou ultérieure.
2. Copiez le fichier <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/.json> dans le répertoire de votre image [amazon-ssm-agent](#). /etc/amazon/ ssm/ Cela active le mode conteneur dans l'agent SSM.

Note

Les images personnalisées nécessiteraient la mise à jour de l'agent SSM le plus récent pour que cette fonctionnalité fonctionne comme prévu.

- Le rôle CodeBuild de service doit respecter la politique SSM suivante :

```
{
  "Effect": "Allow",
  "Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
    "ssmmessages:OpenDataChannel"
  ],
  "Resource": "*"
}
```

Vous pouvez demander à la CodeBuild console d'associer automatiquement cette politique à votre rôle de service lorsque vous démarrez la génération. Vous pouvez également associer cette politique à votre rôle de service manuellement.

- Si l'audit et la journalisation de l'activité des sessions sont activés dans les préférences de Systems Manager, le rôle de CodeBuild service doit également disposer d'autorisations supplémentaires. Les autorisations sont différentes en fonction de l'endroit où les journaux sont stockés.

CloudWatch Journaux

Si vous utilisez CloudWatch Logs pour stocker vos journaux, ajoutez l'autorisation suivante au rôle CodeBuild de service :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:*:*"
    },
    {
```

```
"Effect": "Allow",
"Action": [
  "logs:CreateLogStream",
  "logs:PutLogEvents"
],
"Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:<log-group-
name>:*"
}
]
}
```

Amazon S3

Si vous utilisez Amazon S3 pour stocker vos journaux, ajoutez l'autorisation suivante au rôle CodeBuild de service :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetEncryptionConfiguration",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Pour plus d'informations, consultez la section [Audit et journalisation de l'activité des sessions](#) dans le guide de AWS Systems Manager l'utilisateur.

Suspendre le build

Pour suspendre la génération, insérez la `codebuild-breakpoint` commande dans l'une des phases de construction de votre fichier `buildspec`. La construction sera suspendue à ce stade, ce qui vous

permettra de vous connecter au conteneur de construction et de visualiser le conteneur dans son état actuel.

Par exemple, ajoutez ce qui suit aux phases de construction dans votre fichier buildspec.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Ce code crée le /tmp/hello-world fichier, puis interrompt la compilation à ce stade.

Démarrez la construction

Pour autoriser l'utilisation du gestionnaire de session avec la session de génération, vous devez activer les connexions de session pour la génération. Pour ce faire, lors du démarrage de la compilation, procédez comme suit :

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le volet de navigation, choisissez Projets de génération. Choisissez le projet de construction, puis choisissez Démarrer la construction avec des remplacements.
3. Choisissez Advanced build overrides (Remplacements avancés de génération).
4. Dans la section Environnement, choisissez l'option Activer la connexion à la session. Si cette option n'est pas sélectionnée, toutes les codebuild-resume commandes codebuild-breakpoint et sont ignorées.
5. Apportez les autres modifications souhaitées, puis choisissez Démarrer la construction.
6. Surveillez l'état de construction dans la console. Lorsque la session est disponible, le lien du gestionnaire de AWS session apparaît dans la section État de la création.

Connectez-vous au conteneur de construction

Vous pouvez vous connecter au conteneur de construction de deux manières :

CodeBuild console

Dans un navigateur Web, ouvrez le lien du gestionnaire de AWS session pour vous connecter au conteneur de construction. Une session de terminal s'ouvre pour vous permettre de parcourir et de contrôler le conteneur de construction.

AWS CLI

Note

Le plug-in Session Manager doit être installé sur votre machine locale pour cette procédure. Pour plus d'informations, voir [Installer le plug-in du gestionnaire de session pour la AWS CLI](#) dans le guide de AWS Systems Manager l'utilisateur.

1. Appelez l'batch-get-buildsAPI avec l'ID de build pour obtenir des informations sur le build, y compris l'identifiant cible de session. Le nom de propriété de l'identifiant cible de session varie en fonction du type de sortie de la aws commande. C'est pourquoi --output json il est ajouté à la commande.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Copiez la valeur sessionTarget de la propriété. Le nom de la sessionTarget propriété peut varier en fonction du type de sortie de la aws commande. C'est pourquoi --output json il est ajouté à la commande à l'étape précédente.
3. Utilisez la commande suivante pour vous connecter au conteneur de construction.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Pour cet exemple, vérifiez que le /tmp/hello-world fichier existe et contient le texteHello World.

Reprendre le build

Une fois que vous avez terminé d'examiner le conteneur de construction, émettez la codebuild-resume commande depuis le shell du conteneur.

```
$ codebuild-resume
```

Suppression de générations dans AWS CodeBuild

Vous pouvez utiliser l'AWS CLI ou les kits SDK AWS pour supprimer les générations dans AWS CodeBuild.

Suppression de générations (AWS CLI)

Exécutez la commande `batch-delete-builds` :

```
aws codebuild batch-delete-builds --ids ids
```

Dans la commande précédente, remplacez l'espace réservé suivant :

- *identifiants* : Chaîne obligatoire. ID des générations à supprimer. Pour spécifier plusieurs générations, séparez les ID par un espace. Pour obtenir une liste d'ID de générations, consultez les rubriques suivantes :
 - [Affichage d'une liste d'ID de génération \(AWS CLI\)](#)
 - [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#)

En cas de réussite, un tableau `buildsDeleted` s'affiche dans la sortie avec l'Amazon Resource Name (ARN) de chaque génération supprimée. Des informations sur les générations qui n'ont pas pu être supprimées apparaissent dans la sortie dans un tableau `buildsNotDeleted`.

Par exemple, si vous exécutez la commande suivante :

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Des informations similaires à ce qui suit s'affichent dans la sortie :

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
```

```
"arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-  
project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"  
]  
}
```

Suppression de générations (kits SDK AWS)

Pour de plus amples informations sur l'utilisation d'AWS CodeBuild avec le kit SDK AWS, consultez la [Référence des kits SDK et des outils AWS](#).

Utilisation du AWS Lambda calcul dans AWS CodeBuild

AWS Lambda compute offre des vitesses de démarrage optimisées pour vos builds. AWS Lambda prend en charge des builds plus rapides grâce à une latence de démarrage plus faible. AWS Lambda évolue également automatiquement, de sorte que les builds n'attendent pas dans la file d'attente pour s'exécuter. Cependant, certains cas d'utilisation AWS Lambda ne sont pas compatibles, et s'ils vous concernent, utilisez le calcul EC2. Pour plus d'informations, consultez [Limites du AWS Lambda calcul](#).

Rubriques

- [Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda](#)
- [Dans quelles régions le AWS Lambda calcul est pris en charge CodeBuild ?](#)
- [Limites du AWS Lambda calcul](#)
- [AWS Lambda calculer des échantillons avec AWS CodeBuild](#)

Quels outils et environnements d'exécution seront inclus dans les images docker de l'environnement d'exécution sélectionnées sur lesquelles s'exécutent ? AWS Lambda

AWS Lambda prend en charge les outils suivants : AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby et .NET.

Dans quelles régions le AWS Lambda calcul est pris en charge CodeBuild ?

En CodeBuild, le AWS Lambda calcul est pris en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort), Europe (Irlande) et Amérique du Sud (São Paulo). Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Limites du AWS Lambda calcul

Certains cas d'utilisation AWS Lambda ne sont pas compatibles, et s'ils vous concernent, utilisez le calcul EC2 :

- AWS Lambda ne prend pas en charge les outils qui nécessitent des autorisations root. Pour des outils tels que yum ou rpm, utilisez le type de calcul EC2 ou d'autres outils ne nécessitant pas d'autorisations root.
- AWS Lambda ne prend pas en charge les builds ou les exécutions de Docker. Vous pouvez utiliser des alternatives qui ne nécessitent pas d'autorisations root, telles que Podman.
- AWS Lambda ne prend pas en charge l'écriture dans des fichiers externes/tmp. Les gestionnaires de packages inclus sont configurés pour utiliser le /tmp répertoire par défaut pour le téléchargement et le référencement des packages.
- AWS Lambda ne prend pas en charge le type d'environnement LINUX_GPU_CONTAINER et n'est pas pris en charge sur Windows Server Core 2019.
- AWS Lambda ne prend pas en charge la mise en cache, les compilations par lots, les délais de génération personnalisés, les délais d'attente, les badges de génération, le mode privilégié, les environnements d'exécution personnalisés ou les durées d'exécution supérieures à 15 minutes.
- AWS Lambda ne prend pas en charge la connectivité VPC, une plage fixe d'adresses IP CodeBuild sources, le système EFS, le versionnement sémantique, l'installation de certificats ou l'accès SSH avec le gestionnaire de session.

AWS Lambda calculer des échantillons avec AWS CodeBuild

Ces groupes d'échantillons peuvent être utilisés pour expérimenter le AWS Lambda calcul in CodeBuild.

Rubriques

- [Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java](#)
- [Créez une application React d'une seule page avec CodeBuild Lambda Node.js](#)
- [Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python](#)

Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Java

The AWS Serverless Application Model (AWS SAM) est un framework open source permettant de créer des applications sans serveur. Pour plus d'informations, consultez le [AWS Serverless Application Model référentiel](#) sur GitHub. L'exemple Java suivant utilise Gradle pour créer et tester une AWS Lambda fonction. Ensuite, la AWS SAM CLI est utilisée pour déployer le AWS CloudFormation modèle et le bundle de déploiement. Grâce à CodeBuild Lambda, les étapes de construction, de test et de déploiement sont toutes gérées automatiquement, ce qui permet de mettre à jour rapidement l'infrastructure sans intervention manuelle lors d'une seule génération.

Configurez votre AWS SAM référentiel

Créez un AWS SAM Hello World projet à l'aide de la AWS SAM CLI.

Pour créer votre AWS SAM projet

1. Suivez les instructions du guide du AWS Serverless Application Model développeur pour [installer la AWS SAM CLI](#) sur votre machine locale.
2. Exécutez `sam init` et sélectionnez la configuration de projet suivante.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions? [y/N]: N
Project name [sam-app]: <insert project name>
```

3. Téléchargez le dossier AWS SAM du projet dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Java

Créez un projet AWS CodeBuild Lambda Java et configurez les autorisations IAM nécessaires pour le build.

Pour créer votre projet CodeBuild Lambda Java

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms de projet de génération doivent être uniques dans chaque compte AWS. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.
5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Java.
 - Pour Image, sélectionnez aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21.
 - Pour Rôle de service, laissez l'option Nouveau rôle de service sélectionnée. Notez le nom du rôle. Cela sera nécessaire lorsque vous mettrez à jour les autorisations IAM du projet ultérieurement dans cet exemple.
6. Choisissez Créer un projet de génération.
7. Ouvrez la console IAM à l'[adresse https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
8. Dans le volet de navigation, choisissez Rôles et sélectionnez le rôle de service associé à votre projet. Vous pouvez trouver votre rôle dans le projet en CodeBuild sélectionnant votre projet de construction, en choisissant Modifier, Environnement, puis Rôle de service.
9. Sélectionnez l'onglet Trust relationships (Relations d'approbation), puis Edit trust policy (Modifier la politique d'approbation).
10. Ajoutez la politique intégrée suivante à votre rôle IAM. Cela sera utilisé pour déployer votre AWS SAM infrastructure ultérieurement. Pour plus d'informations, consultez la rubrique [Ajout et suppression d'autorisations basées sur l'identité IAM](#) du Guide de l'utilisateur IAM.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Action": [
      "cloudformation:*",
      "lambda:*",
      "iam:*",
      "apigateway:*",
      "s3:*"
    ],
    "Resource": [
      "*"
    ]
  }
]
```

Configurer le buildspec du projet

Pour créer, tester et déployer votre fonction Lambda, CodeBuild lit et exécute les commandes de construction à partir d'une spécification de construction.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Passer à l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
env:
  variables:
    GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
```

```
- sam build --template-file template.yaml
- echo "Running deploy..."
- sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
- yes | sam deploy
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Déployez votre infrastructure AWS SAM Lambda

Utilisez CodeBuild Lambda pour déployer automatiquement votre infrastructure Lambda

Pour déployer votre infrastructure Lambda

1. Choisissez Démarrer la génération. Cela permettra de créer, de tester et de déployer automatiquement votre AWS SAM application pour AWS Lambda l'utiliserAWS CloudFormation.
2. Une fois le build terminé, accédez à la AWS Lambda console et recherchez votre nouvelle fonction Lambda sous le nom du AWS SAM projet.
3. Testez votre fonction Lambda en sélectionnant API Gateway dans la vue d'ensemble des fonctions, puis en cliquant sur l'URL du point de terminaison de l'API. Vous devriez voir une page s'ouvrir avec le message "message": "hello world".

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées par votre AWS SAM modèle et CodeBuild.

Pour nettoyer votre infrastructure

1. Accédez à la AWS CloudFormation console et sélectionnez leaws-sam-cli-managed-default.
2. Dans Ressources, videz le compartiment de déploiementSamCliSourceBucket.
3. Supprimez la aws-sam-cli-managed-default pile.
4. Supprimez la AWS CloudFormation pile associée à votre AWS SAM projet. Cette pile doit porter le même nom que votre AWS SAM projet.
5. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.

6. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.

Créez une application React d'une seule page avec CodeBuild Lambda Node.js

[Create React App](#) est un moyen de créer des applications React d'une seule page. L'exemple de fichier Node.js suivant utilise Node.js pour créer les artefacts source à partir de Create React App et renvoie les artefacts de construction.

Configuration de votre référentiel source et de votre compartiment d'artefacts

Créez un référentiel source pour votre projet à l'aide de yarn et de Create React App.

Pour configurer le référentiel source et le compartiment d'artefacts

1. Sur votre machine locale, lancez-le `yarn create react-app <app-name>` pour créer une application React simple.
2. Téléchargez le dossier du projet de l'application React dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Node.js

Créez un projet AWS CodeBuild Lambda Node.js.

Pour créer votre projet CodeBuild Lambda Node.js

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Créer un projet de construction. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms de projet de génération doivent être uniques dans chaque compte AWS. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.

5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Node.js.
 - Pour Image, sélectionnez aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.
6. Dans Artefacts:
 - Pour Type, sélectionnez Amazon S3.
 - Dans le champ Nom du compartiment, sélectionnez le compartiment d'artefacts du projet que vous avez créé précédemment.
 - Pour l'emballage des artefacts, sélectionnez Zip.
7. Choisissez Créer un projet de génération.

Configurer le buildspec du projet

Pour créer votre application React, CodeBuild lit et exécute les commandes de construction à partir d'un fichier buildspec.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Basculer vers l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
phases:
  build:
    commands:
      - yarn
      - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
      - yarn run build
      - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
artifacts:
  name: "build-output"
  files:
    - "**/*"
```

```
reports:
  test-report:
    files:
      - 'junit.xml'
    file-format: 'JUNITXML'
  coverage-report:
    files:
      - 'coverage/clover.xml'
    file-format: 'CLOVERXML'
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Créez et exécutez votre application React

Créez l'application React sur CodeBuild Lambda, téléchargez les artefacts de construction et exécutez l'application React localement.

Pour créer et exécuter votre application React

1. Choisissez Démarrer la génération.
2. Une fois le build terminé, accédez au bucket d'artefacts de votre projet Amazon S3 et téléchargez l'artefact de l'application React.
3. Décompressez l'artefact de construction de React et placez-le `run npm install -g serve && serve -s build` dans le dossier du projet.
4. La `serve` commande servira le site statique sur un port local et imprimera le résultat sur votre terminal. Vous pouvez visiter l'URL localhost ci-dessous `Local` : dans la sortie du terminal pour afficher votre application React.

Pour en savoir plus sur la gestion du déploiement d'un serveur basé sur React, voir [Créer un déploiement d'applications React](#).

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées pour votre CodeBuild projet.

Pour nettoyer votre infrastructure

1. Supprimer les artefacts de votre projet (compartiment Amazon S3)

2. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.
3. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.

Mettre à jour une configuration de fonction Lambda avec CodeBuild Lambda Python

L'exemple Python suivant utilise [Boto3](#) et Lambda CodeBuild Python pour mettre à jour la configuration d'une fonction Lambda. Cet exemple peut être étendu pour gérer d'autres AWS ressources par programmation. Pour plus d'informations, consultez la documentation de [Boto3](#).

Prérequis

Créez ou trouvez une fonction Lambda dans votre compte.

Cet exemple suppose que vous avez déjà créé une fonction Lambda dans votre compte et que vous l'utiliserez CodeBuild pour mettre à jour les variables d'environnement de la fonction Lambda. Pour plus d'informations sur la configuration d'une fonction Lambda via CodeBuild, consultez [l'Déployer une fonction Lambda à l'aide de AWS SAM CodeBuild Lambda Javaexemple](#) ou rendez-vous sur [AWS Lambda](#)

Configurez votre référentiel de sources

Créez un dépôt source pour stocker votre script python Boto3.

Pour configurer le référentiel source

1. Copiez le script python suivant dans un nouveau fichier appelé `update_lambda_environment_variables.py`.

```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
```



```
print("Updating lambda function " + lambda_function_name + " environment
variable "
      + lambda_env_variable + " to " + lambda_env_variable_value)
lambda_client.update_function_configuration(
    FunctionName=lambda_function_name,
    Environment={
        'Variables': {
            lambda_env_variable: lambda_env_variable_value
        }
    },
)

if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Téléchargez le fichier python dans un référentiel source compatible. Pour obtenir la liste des types de sources pris en charge, consultez [ProjectSource](#).

Création d'un projet CodeBuild Lambda Python

Créez un projet CodeBuild Lambda Python.

Pour créer votre projet CodeBuild Lambda Java

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si une page CodeBuild d'informations s'affiche, choisissez Create build project. Sinon, dans le volet de navigation, développez Build, choisissez Build projects, puis Create build project.
3. Dans Nom du projet, saisissez un nom pour ce projet de génération. Les noms de projet de génération doivent être uniques dans chaque compte AWS. Vous pouvez également inclure une description facultative du projet de construction pour aider les autres utilisateurs à comprendre à quoi sert ce projet.
4. Dans Source, sélectionnez le référentiel source dans lequel se trouve votre AWS SAM projet.
5. Dans Environment (Environnement) :
 - Pour Compute, sélectionnez Lambda.
 - Pour Runtime (s), sélectionnez Python.

- Pour Image, sélectionnez `aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`.
 - Pour Rôle de service, laissez l'option Nouveau rôle de service sélectionnée. Notez le nom du rôle. Cela sera nécessaire lorsque vous mettrez à jour les autorisations IAM du projet ultérieurement dans cet exemple.
6. Choisissez Créer un projet de génération.
 7. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
 8. Dans le volet de navigation, choisissez Rôles et sélectionnez le rôle de service associé à votre projet. Vous pouvez trouver votre rôle dans le projet en CodeBuild sélectionnant votre projet de construction, en choisissant Modifier, Environnement, puis Rôle de service.
 9. Sélectionnez l'onglet Trust relationships (Relations d'approbation), puis Edit trust policy (Modifier la politique d'approbation).
 10. Ajoutez la politique intégrée suivante à votre rôle IAM. Cela sera utilisé pour déployer votre AWS SAM infrastructure ultérieurement. Pour plus d'informations, consultez la rubrique [Ajout et suppression d'autorisations basées sur l'identité IAM](#) du Guide de l'utilisateur IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateLambdaPermissions",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Configurer le buildspec du projet

Afin de mettre à jour la fonction Lambda, le script lit les variables d'environnement depuis le buildspec pour trouver le nom, le nom de la variable d'environnement et la valeur de la variable d'environnement de la fonction Lambda.

Pour configurer le buildspec de votre projet

1. Dans la CodeBuild console, sélectionnez votre projet de build, puis choisissez Edit et Buildspec.
2. Dans Buildspec, choisissez Insérer des commandes de construction, puis Passer à l'éditeur.
3. Supprimez les commandes de construction préremplies et collez les spécifications de construction suivantes.

```
version: 0.2
env:
  variables:
    LAMBDA_FUNC_NAME: "<lambda-function-name>"
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
    LAMBDA_ENV_VARIABLE_VALUE: "true"
phases:
  install:
    commands:
      - pip3 install boto3
  build:
    commands:
      - python3 update_lambda_environment_variables.py
```

4. Choisissez Update buildspec (Mettre à jour buildspec).

Mettez à jour votre configuration Lambda

Utilisez CodeBuild Lambda Python pour mettre à jour automatiquement la configuration de votre fonction Lambda.

Pour mettre à jour la configuration de votre fonction Lambda

1. Choisissez Démarrer la génération.
2. Une fois le build terminé, accédez à votre fonction Lambda.
3. Sélectionnez Configuration, puis Variables d'environnement. Vous devriez voir une nouvelle variable d'environnement contenant une clé FEATURE_ENABLED et une valeur true.

Nettoyez votre infrastructure

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez les ressources créées pour votre CodeBuild projet.

Pour nettoyer votre infrastructure

1. Accédez à la CloudWatch console et supprimez les groupes de CloudWatch journaux associés à votre CodeBuild projet.
2. Accédez à la CodeBuild console et supprimez votre CodeBuild projet en choisissant Supprimer le projet de construction.
3. Si vous avez créé une fonction Lambda aux fins de cet exemple, choisissez Actions et fonction Supprimer pour nettoyer votre fonction Lambda.

Extensions

Si vous souhaitez étendre cet exemple pour gérer d'autres AWS ressources à l'aide de AWS CodeBuild Lambda Python :

- Mettez à jour le script Python pour modifier les nouvelles ressources à l'aide de Boto3.
- Mettez à jour le rôle IAM associé à votre CodeBuild projet afin d'obtenir des autorisations pour les nouvelles ressources.
- Ajoutez toutes les nouvelles variables d'environnement associées aux nouvelles ressources à vos spécifications de construction.

Travailler avec une capacité réservée dans AWS CodeBuild

CodeBuild propose les flottes de calcul suivantes :

- Flottes à la demande
- Flottes de capacité réservée

Avec des flottes à la demande, CodeBuild fournit le calcul nécessaire à vos builds. Les machines sont détruites une fois la construction terminée. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.

Note

Les flottes à la demande ne sont pas compatibles avec Windows Server 2022.

CodeBuild propose également des flottes de capacités réservées contenant des instances alimentées par Amazon EC2 et gérées par CodeBuild. Avec les flottes de capacité réservée, vous configurez un ensemble d'instances dédiées pour votre environnement de construction. Ces machines restent inactives, prêtes à traiter les builds ou les tests immédiatement et réduisent les durées de construction. Avec des flottes de capacité réservées, vos machines fonctionnent en permanence et continueront d'entraîner des coûts tant qu'elles seront approvisionnées.

Important

Quelle que soit la durée pendant laquelle vous exécutez une instance, les flottes de capacité réservée sont facturées initialement par instance, après quoi des coûts supplémentaires peuvent être associés. Pour de plus amples informations, veuillez consulter <https://aws.amazon.com/codebuild/pricing/>.

Rubriques

- [Comment démarrer avec les flottes à capacité réservée ?](#)
- [Bonnes pratiques](#)
- [Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ?](#)
- [Quelles régions prennent en charge les flottes à capacité réservée ?](#)

- [Propriétés du parc à capacité réservée](#)
- [Échantillons de capacité réservée avec AWS CodeBuild](#)
- [Limitations des flottes à capacité réservée](#)

Comment démarrer avec les flottes à capacité réservée ?

Pour créer un parc de capacités réservées

1. Connectez-vous à la AWS CodeBuild console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Compute fleets, puis Create compute fleet.
3. Dans le champ de texte Calculer le nom du parc, saisissez le nom de votre parc.
4. Dans le menu déroulant Système d'exploitation, sélectionnez le système d'exploitation.
5. Dans le menu déroulant Architecture, sélectionnez l'architecture.
6. Dans le menu déroulant Calculer, choisissez le type de machine de calcul pour votre machine.
7. Dans le champ de texte Capacité, entrez le nombre minimum d'instances dans le parc.
8. Dans le champ Comportement de débordement, choisissez le comportement lorsque la demande dépasse la capacité du parc. Pour plus d'informations sur ces options, consultez [Propriétés du parc à capacité réservée](#).
9. Choisissez Créer un parc informatique.
10. Une fois le parc informatique créé, créez un nouveau CodeBuild projet ou modifiez un projet existant. Dans Environnement, choisissez Capacité réservée sous Modèle de provisionnement, puis choisissez le parc spécifié sous Nom du parc.

Bonnes pratiques

Lorsque vous utilisez des flottes de capacité réservée, nous vous recommandons de suivre ces bonnes pratiques.

- Nous vous recommandons d'utiliser le mode de cache source pour améliorer les performances de génération en mettant en cache la source.
- Nous vous recommandons d'utiliser la mise en cache des couches Docker pour améliorer les performances de création en mettant en cache les couches Docker existantes.

Puis-je partager un parc de capacités réservées entre plusieurs CodeBuild projets ?

Oui, vous pouvez optimiser l'utilisation de la capacité d'une flotte en l'utilisant dans le cadre de plusieurs projets.

Quelles régions prennent en charge les flottes à capacité réservée ?

Les flottes de capacité réservée sont prises en charge dans les pays suivants Régions AWS : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort), Europe (Irlande) et Amérique du Sud (São Paulo). Pour plus d'informations sur Régions AWS les CodeBuild zones disponibles, consultez la section [AWS Services par région](#).

Propriétés du parc à capacité réservée

Un parc à capacité réservée contient les propriétés suivantes :

Système d'exploitation

Le système d'exploitation. Les systèmes d'exploitation suivants sont disponibles :

- Amazon Linux
- Windows Server 2019
- Windows Server 2022

Architecture

Architecture du processeur. Les architectures disponibles sont les suivantes :

- x86_64
- Bras 64

Calcul

Type de machine de calcul pour chaque instance. Les types de machines suivants sont disponibles :

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | vCPU | Espace disque |
|---------------------------|---------------------------------------|--------------------------------|---------|------|---------------|
| ARM Petit | BUILD_GENERAL1_SMALL | ARM_CONTAINER | 4 Go | 2 | 50 Go |
| ARM Grand | BUILD_GENERAL1_LARGE | ARM_CONTAINER | 16 Go | 8 | 50 Go |
| Linux Small ¹ | BUILD_GENERAL1_SMALL | LINUX_CONTAINER | 3 Go | 2 | 64 Go |
| Linux Medium ¹ | BUILD_GENERAL1_MEDIUM | LINUX_CONTAINER | 7 Go | 4 | 128 Go |
| Linux Large ¹ | BUILD_GENERAL1_LARGE | LINUX_CONTAINER | 15 Go | 8 | 128 Go |
| Linux XLarge | BUILD_GENERAL1_XLARGE | LINUX_CONTAINER | 70 GO | 36 | 256 Go |
| Linux 2 x Large | BUILD_GENERAL1_2XLARGE | LINUX_CONTAINER | 145 Go | 72 | 824 Go SSD |
| GPU Linux petit | BUILD_GENERAL1_SMALL | LINUX_GPU_CONTAINER | 16 Go | 4 | 220 GO |

| Type de calcul | Valeur ComputeType de l'environnement | Valeur du type d'environnement | Mémoire | vCPU | Espace disque |
|----------------------------|---------------------------------------|--------------------------------|---------|------|---------------|
| GPU Linux de grande taille | BUILD_GENERAL1_LARGE | LINUX_GPU_CONTAINER | 255 Go | 32 | 50 Go |
| Windows Medium | BUILD_GENERAL1_MEDIUM | WINDOWS_SERVER_2019_CONTAINER | 7 Go | 4 | 128 Go |
| Windows Medium | BUILD_GENERAL1_MEDIUM | WINDOWS_SERVER_2022_CONTAINER | 7 Go | 4 | 128 Go |
| Fenêtres grandes | BUILD_GENERAL1_LARGE | WINDOWS_SERVER_2019_CONTAINER | 15 Go | 8 | 128 Go |
| Fenêtres grandes | BUILD_GENERAL1_LARGE | WINDOWS_SERVER_2022_CONTAINER | 15 Go | 8 | 128 Go |

Capacité

Le nombre initial de machines allouées au parc, qui définit le nombre de builds pouvant être exécutés en parallèle.

Comportement de débordement

Définit le comportement lorsque le nombre de builds dépasse la capacité de la flotte.

À la demande

Les versions Overflow s'exécutent CodeBuild à la demande.

Important

Si vous choisissez de définir votre comportement de débordement à la demande, notez que les versions de débordement seront facturées séparément, comme pour Amazon EC2 à la demande. Pour de plus amples informations, veuillez consulter <https://aws.amazon.com/codebuild/pricing/>.

File d'attente

Les cycles de compilation sont placés dans une file d'attente jusqu'à ce qu'une machine soit disponible. Cela limite les coûts supplémentaires car aucune machine supplémentaire n'est allouée.

Échantillons de capacité réservée avec AWS CodeBuild

Ces échantillons peuvent être utilisés pour expérimenter avec des flottes de capacité réservée dans CodeBuild.

Rubriques

- [Exemple de mise en cache avec capacité réservée](#)

Exemple de mise en cache avec capacité réservée

Un cache peut stocker les éléments réutilisables de votre environnement de génération et les utiliser sur plusieurs générations. Cet exemple montre comment activer la mise en cache dans votre projet de construction en utilisant une capacité réservée. Pour de plus amples informations, veuillez consulter [Mise en cache de génération dans AWS CodeBuild](#).


Vous pouvez commencer par spécifier un ou plusieurs modes de cache dans les paramètres de votre projet :

Cache :

Type: LOCAL

Modes :

- LOCAL_CUSTOM_CACHE
- LOCAL_DOCKER_LAYER_CACHE
- LOCAL_SOURCE_CACHE

 Note

Assurez-vous d'activer le mode privilégié afin d'utiliser le cache de couche Docker.

Les paramètres buildspec de votre projet doivent ressembler à ce qui suit :

```
version: 0.2
  phases:
    build:
      commands:
        - echo testing local source cache
        - touch /codebuild/cache/workspace/foobar.txt
        - git checkout -b cached_branch
        - echo testing local docker layer cache
        - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
        - echo testing local custom cache
        - touch foo
        - mkdir bar && ln -s foo bar/foo2
        - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
        - "[ -f foo ] || exit 1"
        - "[ -L bar/foo2 ] || exit 1"
        - "[ -f bar/bar/foo3 ] || exit 1"
        - "[ -f bar/bar/foo4 ] || exit 1"
      cache:
        paths:
          - './foo'
          - './bar/**/*'
          - './bar/bar/foo3'
```

Vous pouvez commencer par exécuter une compilation avec le nouveau projet pour démarrer le cache. Une fois cette opération terminée, vous devez commencer une autre version avec une spécification de construction alternative, similaire à ce qui suit :

```
version: 0.2
  phases:
```

```
build:
  commands:
    - echo testing local source cache
    - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
    - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
    - echo testing local docker layer cache
    - docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
    - echo testing local custom cache
    - "[ -f foo ] || exit 1"
    - "[ -L bar/foo2 ] || exit 1"
    - "[ -f bar/bar/foo3 ] || exit 1"
    - "[ -f bar/bar/foo4 ] || exit 1"
  cache:
    paths:
      - './foo'
      - './bar/**/*'
      - './bar/bar/foo3'
```

Limitations des flottes à capacité réservée

Il existe certains cas d'utilisation que les flottes de capacité réservée ne prennent pas en charge, et s'ils vous concernent, utilisez plutôt des flottes à la demande :

- Les flottes de capacités réservées ne prennent pas en charge les builds par lots, les métriques d'utilisation des builds ou le versionnement sémantique.
- Les flottes de capacité réservée ne prennent pas en charge la connectivité VPC.

Pour plus d'informations sur les limites et les quotas, consultez [Flottes informatiques](#).

Utilisation des rapports de test dans AWS CodeBuild

Vous pouvez créer des rapports CodeBuild contenant des détails sur les tests exécutés pendant les builds. Vous pouvez créer des tests tels que des tests unitaires, des tests de configuration et des tests fonctionnels.

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

- Concombre JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- Unit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)

Note

La dernière version prise en charge de `cucumber-js` est la 7.3.2.

Créez vos cas de test avec n'importe quel framework de test capable de créer des fichiers de rapport dans l'un de ces formats (par exemple, le plug-in Surefire JUnit, TestNG et Cucumber).

Pour créer un rapport de test, ajoutez un nom de groupe de rapports au fichier buildspec d'un projet de construction avec des informations sur vos cas de test. Lorsque vous exécutez le projet de construction, les cas de test sont exécutés et un rapport de test est créé. Vous n'avez pas besoin de créer un groupe de rapports avant d'exécuter vos tests. Si vous spécifiez un nom de groupe de rapports, il en CodeBuild crée un pour vous lorsque vous générez vos rapports. Si vous souhaitez utiliser un groupe de rapports qui existe déjà, vous spécifiez son ARN dans le fichier buildspec.

Vous pouvez utiliser un rapport de test pour résoudre un problème lors d'une génération. Si vous disposez de nombreux rapports de test provenant de plusieurs versions d'un projet de construction, vous pouvez utiliser vos rapports de test pour afficher les tendances et les taux de test et d'échec afin de vous aider à optimiser les versions.

Un rapport expire 30 jours après sa création. Vous ne pouvez pas afficher un rapport de test expiré. Si vous souhaitez conserver les rapports de test pendant plus de 30 jours, vous pouvez exporter les

fichiers de données brutes de vos résultats de test vers un compartiment Amazon S3. Les fichiers de test exportés n'expirent pas. Des informations sur le compartiment S3 sont spécifiées lorsque vous créez le groupe de rapports.

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Rubriques

- [Créer un rapport de test](#)
- [Utilisation de groupes de rapports](#)
- [Utilisation des rapports](#)
- [Utilisation des autorisations de rapport de test](#)
- [Affichage des rapports de test](#)
- [Rapport de test avec des cadres de test](#)
- [Rapports de couverture du code](#)

Créer un rapport de test

Pour créer un rapport de test, vous exécutez un projet de construction qui est configuré avec un à cinq groupes de rapports dans son fichier buildspec. Un rapport de test est créé pendant l'exécution. Il contient les résultats des cas de test spécifiés pour les groupes de rapports. Un nouveau rapport de test est généré pour chaque génération ultérieure qui utilise le même fichier buildspec.

Pour créer un rapport de test

1. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération dans AWS CodeBuild](#).
2. Configurez le fichier buildspec de votre projet avec les informations de rapport de test :
 - a. Ajoutez une `reports` : section et spécifiez soit l'ARN d'un groupe de rapports existant, soit le nom d'un groupe de rapports.

Si vous spécifiez un ARN, CodeBuild utilise ce groupe de rapports.

Si vous spécifiez un nom, CodeBuild crée un groupe de rapports pour vous en utilisant le nom de votre projet et le nom que vous avez spécifié, au format `<project-name>- <report-group-name >`. Si le groupe de rapports nommé existe déjà, CodeBuild utilise ce groupe de rapports.

- b. Dans le groupe de rapports, spécifiez l'emplacement des fichiers contenant les résultats des tests. Si vous utilisez plusieurs groupes de rapports, spécifiez les emplacements des fichiers de résultats de test pour chacun d'eux. Un nouveau rapport de test est créé chaque fois que votre projet de génération s'exécute. Pour de plus amples informations, veuillez consulter [Spécification des fichiers de test](#).
- c. Dans la section `commands` de la séquence `build` ou `post_build`, spécifiez les commandes qui exécutent les cas de test que vous avez spécifiés pour vos groupes de rapports. Pour de plus amples informations, veuillez consulter [Spécification des commandes de test](#).

Voici un exemple de section `buildspec reports` :

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Exécutez une version du projet de génération. Pour de plus amples informations, veuillez consulter [Exécution d'une génération dans AWS CodeBuild](#).
4. Une fois la génération terminée, choisissez la nouvelle version dans Historique de génération sur la page de votre projet. Choisissez Rapports pour afficher le rapport de test. Pour de plus amples informations, veuillez consulter [Affichage des rapports de test d'une génération](#).

Utilisation de groupes de rapports

Un groupe de rapports contient des rapports de test et spécifie des paramètres partagés. Vous utilisez le fichier `buildspec` pour spécifier les cas de test à exécuter et les commandes pour les

exécuter lors de la génération. Pour chaque groupe de rapports configuré dans un projet de génération, une exécution du projet de génération crée un rapport de test. Plusieurs exécutions d'un projet de génération configurées avec un groupe de rapports créent plusieurs rapports de test dans ce groupe de rapports, chacun avec les résultats des mêmes cas de test spécifiés pour ce groupe de rapports.

Les cas de test sont spécifiés pour un groupe de rapports dans le fichier buildspec d'un projet de génération. Vous pouvez spécifier jusqu'à cinq groupes de rapports dans un projet de génération. Lorsque vous exécutez une génération, tous les cas de test s'exécutent. Un nouveau rapport de test est créé avec les résultats de chaque cas de test spécifié pour un groupe de rapports. Chaque fois que vous exécutez une nouvelle version, les cas de test s'exécutent et un nouveau rapport de test est créé avec les nouveaux résultats de test.

Les groupes de rapports peuvent être utilisés dans plusieurs projets de génération. Tous les rapports de test créés avec un groupe de rapports partagent la même configuration, telle que son option d'exportation et ses autorisations, même si les rapports de test sont créés à l'aide de différents projets de génération. Les rapports de test créés avec un groupe de rapports dans plusieurs projets de génération peuvent contenir les résultats de l'exécution de différents ensembles de cas de test (un ensemble de cas de test pour chaque projet de génération). En effet, vous pouvez spécifier différents fichiers de cas de test pour le groupe de rapports dans le fichier buildspec de chaque projet. Vous pouvez également modifier les fichiers de cas de test d'un groupe de rapports dans un projet de génération en modifiant son fichier buildspec. Les versions ultérieures créent de nouveaux rapports de test qui contiennent les résultats des fichiers de cas de test dans le buildspec mis à jour.

Rubriques

- [Créer un groupe de rapports](#)
- [Mise à jour d'un groupe de rapports](#)
- [Spécification des fichiers de test](#)
- [Spécification des commandes de test](#)
- [Attribution des noms des groupes de rapports](#)
- [Balisage des groupes de rapports dans AWS CodeBuild](#)
- [Utilisation des groupes de rapports partagés.](#)

Créer un groupe de rapports

Vous pouvez utiliser la CodeBuild console AWS CLI, le fichier ou un fichier buildspec pour créer un groupe de rapports. Votre rôle IAM doit disposer des autorisations requises pour créer un groupe de rapports. Pour de plus amples informations, veuillez consulter [Utilisation des autorisations de rapport de test](#).

Rubriques

- [Création d'un groupe de rapports \(buildspec\)](#)
- [Créer un groupe de rapports \(console\)](#)
- [Création d'un groupe de rapports \(CLI\)](#)
- [Création d'un groupe de rapports \(AWS CloudFormation\)](#)

Création d'un groupe de rapports (buildspec)

Un groupe de rapports créé à l'aide de buildspec n'exporte pas les fichiers de résultats de test bruts. Vous pouvez afficher votre groupe de rapports et spécifier les paramètres d'exportation. Pour de plus amples informations, veuillez consulter [Mise à jour d'un groupe de rapports](#).

Pour créer un groupe de rapports à l'aide d'un fichier buildspec

1. Choisissez un nom de groupe de rapports qui n'est pas associé à un groupe de rapports dans votre AWS compte.
2. Configurez la section `reports` du fichier buildspec avec ce nom. Dans cet exemple, le nom du groupe de rapports est `new-report-group` et les cas de test d'utilisation sont créés avec le framework JUnit :

```
reports:
  new-report-group: #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Le nom du groupe de rapports peut également être spécifié à l'aide de variables d'environnement dans le buildspec :

```
version: 0.2
env:
```

```
variables:
  REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
...
reports:
  $REPORT_GROUP_NAME:
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Pour plus d'informations, consultez [Spécification des fichiers de test](#) et [Reports syntax in the buildspec file](#).

3. Dans la section `commands`, spécifiez la commande pour exécuter vos tests. Pour de plus amples informations, veuillez consulter [Spécification des commandes de test](#).
4. Exécutez la génération. Une fois la génération terminée, un nouveau groupe de rapports est créé avec un nom qui utilise le format `project-name-report-group-name`. Pour de plus amples informations, veuillez consulter [Attribution des noms des groupes de rapports](#).


Créer un groupe de rapports (console)

Pour créer un rapport de test

1. Ouverture d'AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez Créer un groupe de rapports.
4. Pour Nom du groupe de rapports, entrez un nom pour votre groupe de rapports.
5. (Facultatif) Pour Balises, saisissez le nom et la valeur des balises que vous souhaitez que les services AWS de prise en charge utilisent. Utilisez Ajouter une ligne pour ajouter une balise. Vous pouvez ajouter jusqu'à 50 balises.
6. Si vous souhaitez télécharger les données brutes des résultats de votre rapport de test dans un compartiment Amazon S3 :
 - a. Tâche de sélection Exporter vers Amazon S3.
 - b. Pour le nom du compartiment S3, entrez le nom du compartiment S3.

- c. (Facultatif) Pour Propriétaire du compartiment S3, entrez le AWS identifiant de compte du compte propriétaire du compartiment S3. Cela permet de signaler les données à exporter vers un compartiment Amazon S3 appartenant à un compte autre que celui exécutant la build.
- d. Pour Préfixe du chemin d'accès, entrez le chemin de votre compartiment S3 où vous souhaitez télécharger les résultats de vos tests.
- e. Sélectionnez Compresser les données de résultats de test dans un fichier zip pour compresser vos fichiers de données de résultats de test bruts.
- f. Développez Configuration supplémentaire pour afficher les options de chiffrement. Choisissez l'une des méthodes suivantes :
 - Par défaut AWS Clé gérée par pour utiliser un Clé gérée par AWS pour Amazon S3. Pour de plus amples informations, veuillez consulter [Clés CMK gérées par le client](#) dans le Guide de l'utilisateur AWS Key Management Service. Il s'agit de l'option de chiffrement par défaut.
 - Choisissez une clé personnalisée pour utiliser une clé gérée par le client que vous créez et configurez. Pour la clé de AWS KMS chiffrement, entrez l'ARN de votre clé de chiffrement. Son format est `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-id>`. Pour de plus amples informations, veuillez consulter [Création de clés KMS](#) dans le Guide de l'utilisateur AWS Key Management Service.
 - Désactivez le chiffrement des artefacts pour désactiver le chiffrement. Vous pouvez choisir cette option si vous souhaitez partager vos résultats de test ou les publier sur un site web statique. (Un site web dynamique peut exécuter du code pour déchiffrer les résultats des tests.)

Pour de plus amples informations sur le chiffrement au repos, veuillez consulter [Chiffrement des données](#).

 Note

Le rôle de service CodeBuild spécifié dans le projet est utilisé pour les autorisations de chargement vers le compartiment S3.

7. Choisissez Créer un groupe de rapports.

Création d'un groupe de rapports (CLI)

Pour créer un groupe de rapports

1. Créez un fichier nommé `CreateReportGroup.json`.
2. En fonction de vos besoins, copiez l'un des extraits de code JSON suivants dans `CreateReportGroup.json` :
 - Utilisez le JSON suivant pour spécifier que votre groupe de rapports de test exporte les fichiers de résultats de test bruts vers un compartiment Amazon S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    },
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```

- Remplacez `<bucket-name>` avec le nom de votre compartiment Amazon S3 et `<path>` avec le chemin d'accès de votre compartiment vers l'endroit où vous souhaitez exporter les fichiers.
- Si vous souhaitez compresser les fichiers exportés, pour `packaging`, spécifiez ZIP. Sinon, spécifiez NONE.
- `bucketOwner` est facultatif et n'est obligatoire que si le compartiment Amazon S3 appartient à un compte autre que celui exécutant la build.

- Utilisez `encryptionDisabled` de spécifier si les fichiers exportés doivent être chiffrés. Si vous chiffrez les fichiers exportés, entrez votre clé gérée par le client. Pour plus d'informations, consultez [Mise à jour d'un groupe de rapports](#).
- Utilisez le JSON suivant pour spécifier que votre rapport de test n'exporte pas les fichiers de test bruts :

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

Note

Le rôle de service CodeBuild spécifié dans le projet est utilisé pour les autorisations de chargement vers le compartiment S3.

3. Exécutez la commande suivante :

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

Création d'un groupe de rapports (AWS CloudFormation)

Pour créer un rapport de test à l'aide du modèle AWS CloudFormation

Vous pouvez utiliser un fichier AWS CloudFormation modèle pour créer et configurer un groupe de rapports. Pour plus d'informations, consultez le [AWS CloudFormation guide de l'utilisateur](#).

Le modèle AWS CloudFormation YAML suivant crée un groupe de rapports qui n'exporte pas les fichiers de résultats de test bruts.

Resources:

```
CodeBuildReportGroup:
  Type: AWS::CodeBuild::ReportGroup
  Properties:
```

```
Name: my-report-group-name
Type: TEST
ExportConfig:
  ExportConfigType: NO_EXPORT
```

Le modèle AWS CloudFormation YAML suivant crée un groupe de rapports qui exporte les fichiers de résultats de test bruts vers un compartiment Amazon S3.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: S3
        S3Destination:
          Bucket: my-s3-bucket-name
          Path: path-to-folder-for-exported-files
          Packaging: ZIP
          EncryptionKey: my-KMS-encryption-key
          EncryptionDisabled: false
```

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Mise à jour d'un groupe de rapports

Lorsque vous mettez à jour un groupe de rapports, vous pouvez indiquer s'il convient d'exporter les données brutes des résultats de test vers des fichiers d'un compartiment Amazon S3. Si vous choisissez d'exporter vers un compartiment S3, vous pouvez spécifier les éléments suivants pour votre groupe de rapports :

- Indique si les fichiers de résultats de test bruts sont compressés dans un fichier ZIP.
- Indique si les fichiers de résultats de test bruts sont chiffrés. Vous pouvez spécifier le chiffrement avec l'une des options suivantes :
 - Et Clé gérée par AWS pour Amazon S3.

- Une clé gérée par le client que vous créez et configurez.

Pour de plus amples informations, veuillez consulter [Chiffrement des données](#).

Si vous utilisez le AWS CLI pour mettre à jour un groupe de rapports, vous pouvez également mettre à jour ou ajouter des balises. Pour de plus amples informations, veuillez consulter [Balisage des groupes de rapports dans AWS CodeBuild](#).

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Rubriques

- [Mise à jour d'un groupe de rapports \(console\)](#)
- [Mise à jour d'un groupe de rapports \(CLI\)](#)

Mise à jour d'un groupe de rapports (console)

Pour mettre à jour un groupe de rapports

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le groupe de rapports que vous souhaitez mettre à jour.
4. Choisissez Modifier.
5. Sélectionnez ou désactivez Backup to Amazon S3. Si vous avez sélectionné cette option, spécifiez vos paramètres d'exportation :
 - a. Pour le nom du compartiment S3, entrez le nom du compartiment S3.
 - b. Pour Préfixe du chemin d'accès, entrez le chemin de votre compartiment S3 où vous souhaitez télécharger les résultats de vos tests.
 - c. Sélectionnez Compresser les données de résultats de test dans un fichier zip pour compresser vos fichiers de données de résultats de test bruts.

- d. Développez Configuration supplémentaire pour afficher les options de chiffrement. Sélectionnez l'une des méthodes suivantes :
- Clé AWS gérée par défaut à utiliser Clé gérée par AWS pour Amazon S3. Pour de plus amples informations, veuillez consulter [Clés CMK gérées par le client](#) dans le Guide de l'utilisateur AWS Key Management Service . Il s'agit de l'option de chiffrement par défaut.
 - Choisissez une clé personnalisée pour utiliser une clé gérée par le client que vous créez et configurez. Pour la clé de AWS KMS chiffrement, entrez l'ARN de votre clé de chiffrement. Son format est `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Pour de plus amples informations, veuillez consulter [Création de clés KMS](#) dans le Guide de l'utilisateur AWS Key Management Service .
 - Désactivez le chiffrement des artefacts pour désactiver le chiffrement. Vous pouvez choisir cette option si vous souhaitez partager vos résultats de test ou les publier sur un site web statique. (Un site web dynamique peut exécuter du code pour déchiffrer les résultats des tests.)

Mise à jour d'un groupe de rapports (CLI)

Pour mettre à jour un groupe de rapports

1. Créez un fichier nommé `UpdateReportGroupInput.json`.
2. Copiez ce qui suit dans `UpdateReportGroupInput.json` :

```
{
  "arn": "",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "bucket-name",
      "path": "path",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "your-key"
    }
  },
  "tags": [
    {
      "key": "tag-key",
      "value": "tag-value"
    }
  ]
}
```



```
    }  
  ]  
}
```

- Entrez l'ARN de votre groupe de rapports dans la ligne `arn` (par exemple, `"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1"`)).
- Mettez à jour `UpdateReportGroupInput.json` avec les mises à jour que vous souhaitez appliquer à votre groupe de rapports.
 - Si vous souhaitez mettre à jour votre groupe de rapports pour exporter les fichiers de résultats de test bruts vers un compartiment S3, mettez à jour la section `exportConfig`. Remplacez `bucket-name` par le nom de votre compartiment S3 et `path` par le chemin d'accès de votre compartiment S3 vers l'endroit où vous souhaitez exporter les fichiers. Si vous souhaitez compresser les fichiers exportés, pour `packaging`, spécifiez `ZIP`. Sinon, spécifiez `NONE`. Utilisez `encryptionDisabled` de spécifier si les fichiers exportés doivent être chiffrés. Si vous cryptez les fichiers exportés, entrez votre clé gérée par le client.
 - Si vous souhaitez mettre à jour votre groupe de rapports afin qu'il n'exporte pas les fichiers de résultats de test bruts vers un compartiment S3, mettez à jour la section `exportConfig` avec le code JSON suivant :

```
{  
  "exportConfig": {  
    "exportConfigType": "NO_EXPORT"  
  }  
}
```

- Si vous souhaitez mettre à jour les balises du groupe de rapports, mettez à jour la section `tags`. Vous pouvez modifier, ajouter ou supprimer des balises. Si vous voulez supprimer toutes les balises, mettez-le à jour avec le JSON suivant :

```
"tags": []
```

- Exécutez la commande suivante :

```
aws codebuild update-report-group \  
--cli-input-json file://UpdateReportGroupInput.json
```

Spécification des fichiers de test

Vous spécifiez les fichiers de résultats de test et leur emplacement pour chaque groupe de rapports dans la section `reports` du fichier `buildspec` de votre projet de génération. Pour de plus amples informations, veuillez consulter [Reports syntax in the buildspec file](#).

Voici un exemple de section `reports` qui spécifie deux groupes de rapports pour un projet de génération. L'un est spécifié avec son ARN, l'autre avec un nom. La section `files` spécifie les fichiers qui contiennent les résultats de cas de test. La section facultative `base-directory` spécifie le répertoire où se trouvent les fichiers de cas de test. La section facultative `discard-paths` indique si les chemins d'accès aux fichiers de résultats de test chargés dans un compartiment Amazon S3 sont supprimés.

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
#surefire junit reports
  files:
    - '**/*'
  base-directory: 'surefire/target/surefire-reports'
  discard-paths: false

sampleReportGroup: #Cucumber reports from json plugin
  files:
    - 'cucumber-json/target/cucumber-json-report.json'
  file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Spécification des commandes de test

Vous spécifiez les commandes qui exécutent vos cas de test dans la section `commands` de votre fichier `buildspec`. Ces commandes exécutent les cas de test spécifiés pour vos groupes de rapports dans la section `reports` de votre fichier `buildspec`. Voici un exemple de section `commands` qui inclut des commandes pour exécuter les tests dans les fichiers de test :

```
commands:
  - echo Running tests for surefire junit
  - mvn test -f surefire/pom.xml -fn
  - echo
  - echo Running tests for cucumber with json plugin
  - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
cucumber-json/pom.xml -fn
```

Pour de plus amples informations, veuillez consulter [Syntaxe d'un fichier buildspec](#).

Attribution des noms des groupes de rapports

Lorsque vous utilisez la console AWS CLI ou la AWS CodeBuild console pour créer un groupe de rapports, vous spécifiez un nom pour le groupe de rapports. Si vous utilisez la spécification de build pour créer un nouveau groupe de rapports, il est nommé en utilisant le format *project-name-report-group-name-specified-in-buildspec*. Tous les rapports créés en exécutant des versions de ce projet de génération appartiennent au nouveau groupe de rapports portant le nouveau nom.

Si vous ne souhaitez pas CodeBuild créer de nouveau groupe de rapports, spécifiez l'ARN du groupe de rapports dans le fichier buildspec d'un projet de construction. Vous pouvez spécifier l'ARN d'un groupe de rapports dans plusieurs projets de génération. Après l'exécution de chaque projet de génération, le groupe de rapports contient des rapports de test créés par chaque projet de génération.

Par exemple, si vous créez un groupe de rapports portant le nom `my-report-group`, puis que vous utilisez son nom dans deux projets de génération différents nommés `my-project-1` et `my-project-2` et que vous créez une version des deux projets, deux nouveaux groupes de rapports sont créés. Le résultat est trois groupes de rapports portant les noms suivants :

- `my-report-group` : n'a pas de rapport de test.
- `my-project-1-my-report-group` : contient des rapports avec les résultats des tests exécutés par le projet de génération nommé `my-project-1`.
- `my-project-2-my-report-group` : contient des rapports avec les résultats des tests exécutés par le projet de génération nommé `my-project-2`.

Si vous utilisez l'ARN du groupe de rapports nommé `my-report-group` dans les deux projets, puis que vous exécutez des versions de chaque projet, vous disposez toujours d'un groupe de rapports (`my-report-group`). Ce groupe de rapports contient des rapports de test avec les résultats des tests exécutés par les deux projets de génération.

Si vous choisissez un nom de groupe de rapports qui n'appartient pas à un groupe de rapports dans votre compte AWS , puis que vous utilisez ce nom pour un groupe de rapports dans un fichier buildspec et exécutez une génération de son projet de génération, un nouveau groupe de rapports est créé. Le format du nom du nouveau groupe de rapports est *project-name-new-group-name*. Par exemple, si aucun groupe de rapports ne porte ce nom `new-report-group` dans votre AWS

compte et que vous le spécifiez dans un projet de génération appelé `test-project`, une exécution de génération crée un nouveau groupe de rapports portant ce nom `test-project-new-report-group`.

Balisage des groupes de rapports dans AWS CodeBuild

Une balise est un attribut personnalisé que vous attribuez ou qu'AWS attribue à une ressource AWS. Chaque balise AWS se compose de deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment`, `Project` ou `Secret`). Les clés de balises sont sensibles à la casse.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333`, `Production` ou le nom d'une équipe). Si la valeur de balise est identique à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Ces informations sont qualifiées de paires clé-valeur. Pour les limites sur le nombre de balises que vous pouvez avoir dans un groupe de rapports et les restrictions sur les valeurs et clés de balise, consultez [Balises](#).

Les balises vous aident à identifier et organiser vos ressources AWS. De nombreux services AWS prennent en charge le balisage. Vous pouvez donc attribuer la même balise à des ressources à partir de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à un rapport CodeBuild que celle que vous affectez à un compartiment Amazon S3. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc sur les [bonnes pratiques de balisage](#).

Dans CodeBuild, les ressources principales sont le groupe de rapports et le projet. Vous pouvez utiliser la console CodeBuild, l'AWS CLI, l'API CodeBuild, ou l'AWSSDK pour ajouter, gérer et supprimer des balises pour un groupe de rapports. Outre l'identification, l'organisation et le suivi de votre groupe de rapports avec des balises, vous pouvez utiliser des balises dans les stratégies IAM afin de contrôler qui peut consulter et interagir avec votre groupe de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Rubriques

- [Ajouter une balise à un groupe de rapports](#)
- [Afficher les balises d'un groupe de rapports](#)

- [Modifier les balises d'un groupe de rapports](#)
- [Supprimer une balise d'un groupe de rapports](#)

Ajouter une balise à un groupe de rapports

L'ajout de balises à un groupe de rapports peut vous aider à identifier et organiser vos ressources AWS et à gérer leur accès. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à un groupe de rapports. N'oubliez pas qu'il y a des limites sur le nombre de balises que vous pouvez avoir dans un groupe de rapports. Il existe des restrictions sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour plus d'informations, consultez [Balises](#). Une fois que vous avez des balises, vous pouvez créer des stratégies IAM pour gérer l'accès au groupe de rapports en fonction de ces balises. Vous pouvez utiliser la console CodeBuild ou l'AWS CLI pour ajouter des balises à un groupe de rapports.

Important

L'ajout de balises à un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant d'ajouter une balise à un groupe de rapports, assurez-vous de passer en revue toutes les stratégies IAM qui peuvent utiliser des balises pour contrôler l'accès aux ressources, telles que les groupes de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Pour plus d'informations sur l'ajout de balises à un groupe de rapports lorsque vous le créez, consultez [Créer un groupe de rapports \(console\)](#).

Rubriques

- [Ajouter une balise à un groupe de rapports \(console\)](#)
- [Ajouter une balise à un groupe de rapports \(AWS CLI\)](#)

Ajouter une balise à un groupe de rapports (console)

Vous pouvez utiliser la console CodeBuild pour ajouter une ou plusieurs balises à un groupe de rapports CodeBuild.

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.

2. Dans Report groups (Groupes de rapports), choisissez le nom du groupe de rapports auquel vous souhaitez ajouter des balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Si aucune des balises n'a été ajoutée au groupe de rapports, choisissez Add tag (Ajouter une balise). Vous pouvez également choisir Modifier, puis Ajouter une balise.
5. Dans Key (Clé), entrez un nom de balise. Vous pouvez ajouter une valeur facultative pour la balise dans Value (Valeur).
6. (Facultatif) Pour ajouter une autre balise, choisissez à nouveau Add tag (Ajouter une balise).
7. Lorsque vous avez fini d'ajouter des balises, choisissez Submit (Envoyer).

Ajouter une balise à un groupe de rapports (AWS CLI)

Pour ajouter une balise à un groupe de rapports lorsque vous le créez, veuillez consulter [Création d'un groupe de rapports \(CLI\)](#). Dans `CreateReportGroup.json`, ajoutez vos balises.

Pour ajouter des balises à un groupe de rapports existant, veuillez consulter [Mise à jour d'un groupe de rapports \(CLI\)](#) et ajoutez vos balises dans `UpdateReportGroupInput.json`.

Dans ces étapes, nous supposons que vous avez déjà installé une version récente de l'AWS CLI ou que vous avez procédé à une mise à jour vers la version actuelle. Pour plus d'informations, consultez [Installing the AWS Command Line Interface \(Installation de\)](#).

Afficher les balises d'un groupe de rapports

Les balises peuvent vous aider à identifier et organiser vos ressources AWS et à gérer leur accès. Pour de plus amples informations sur l'utilisation de balises, veuillez consulter le livre blanc sur les [bonnes pratiques de balisage](#). Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Deny or allow actions on report groups based on resource tags](#).

Afficher les balises d'un groupe de rapports (console)

Vous pouvez utiliser la console CodeBuild pour afficher les balises associées à un groupe de rapports CodeBuild.

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dans lequel vous souhaitez afficher les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).

Afficher les balises d'un groupe de rapports (AWS CLI)

Suivez ces étapes pour utiliser l'AWS CLI pour afficher les balises AWS d'un groupe de rapports. Si aucune balise n'a été ajoutée, la liste des balises renvoyées est vide.

1. Utilisez la console ou l'AWS CLI pour localiser l'ARN de votre groupe de rapports. Prenez-en note.

AWS CLI

Exécutez la commande suivante .

```
aws list-report-groups
```

Cette commande renvoie des informations au format JSON semblables à ce qui suit :

```
{
  "reportGroups": [
    "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
  ]
}
```

Un ARN de groupe de rapports se termine par son nom, que vous pouvez utiliser pour identifier l'ARN de votre groupe de rapports.

Console

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.
 2. Dans Groupes de rapports, choisissez le nom de votre groupe de rapports avec les balises que vous souhaitez afficher.
 3. Dans Configuration recherchez l'ARN de votre groupe de rapports.
2. Exécutez la commande suivante . Utilisez l'ARN dont vous avez fait une note pour le paramètre `--report-group-arns`.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

En cas de succès, cette commande renvoie des informations au format JSON qui contiennent une section `tags` similaire à la suivante :

```
{
  ...
  "tags": {
    "Status": "Secret",
    "Project": "TestBuild"
  }
  ...
}
```

Modifier les balises d'un groupe de rapports

Vous pouvez modifier la valeur d'une balise associée à un groupe de rapports. Vous pouvez également modifier le nom de la clé, ce qui équivaut à supprimer la balise et à ajoutant une carte différente avec le nouveau nom et la même valeur que l'autre clé. N'oubliez pas qu'il y a des limites sur les caractères que vous pouvez utiliser dans les champs clé et valeur. Pour plus d'informations, consultez [Balises](#).

Important

La modification des balises d'un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant de modifier le nom (clé) ou la valeur d'une balise pour un groupe de rapports, assurez-vous de passer en revue toutes les stratégies IAM qui peuvent utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources, telles que les groupes de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Deny or allow actions on report groups based on resource tags](#).

Modifier une balise pour un groupe de rapports (console)

Vous pouvez utiliser la console CodeBuild pour modifier les balises associées à un groupe de rapports CodeBuild.

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.
2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dont vous souhaitez modifier les balises.

3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Choisissez Edit (Modifier).
5. Effectuez l'une des actions suivantes :
 - Pour modifier la balise, entrez un nouveau nom dans Key (Clé). La modification du nom de la balise est l'équivalent de la suppression d'une balise et de l'ajout d'une nouvelle balise avec le nouveau nom de clé.
 - Pour modifier la valeur d'une balise, saisissez une nouvelle valeur. Si vous souhaitez modifier la valeur en valeur nulle, supprimez la valeur actuelle et laissez le champ vide.
6. Lorsque vous avez terminé de modifier des balises, choisissez Submit (Soumettre).

Modifier les balises d'un groupe de rapports (AWS CLI)

Pour ajouter, modifier ou supprimer des balises d'un groupe de rapports, veuillez consulter [Mise à jour d'un groupe de rapports \(CLI\)](#). Mettez à jour les balises dans `UpdateReportGroupInput.json`.

Supprimer une balise d'un groupe de rapports

Vous pouvez supprimer une ou plusieurs balises associées à un groupe de rapports. La suppression d'une balise ne supprime pas la balise d'autres ressources AWS qui sont associées à cette balise.

Important

La suppression de balises pour un groupe de rapports peut avoir un impact sur l'accès au groupe de rapports. Avant de supprimer une balise d'un groupe de rapports, assurez-vous de passer en revue toutes les stratégies IAM qui peuvent utiliser la clé ou la valeur d'une balise pour contrôler l'accès aux ressources, telles que les groupes de rapports. Pour obtenir des exemples de stratégies d'accès basées sur les balises, consultez [Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild](#).

Supprimer une balise d'un groupe de rapports (console)

Vous pouvez utiliser la console CodeBuild pour supprimer l'association entre une balise et un groupe de rapports CodeBuild.

1. Ouvrez la console CodeBuild à l'adresse <https://console.aws.amazon.com/codebuild/>.

2. Dans Groupes de rapports, choisissez le nom du groupe de rapports dans lequel vous souhaitez supprimer les balises.
3. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
4. Choisissez Edit (Modifier).
5. Trouvez la balise que vous voulez supprimer, puis choisissez Remove tag (Supprimer une balise).
6. Lorsque vous avez terminé de supprimer les balises, choisissez Submit (Envoyer).

Supprimer une balise d'un groupe de rapports (AWS CLI)

Procédez comme suit pour utiliser l'AWS CLI pour supprimer une balise d'un groupe de rapports CodeBuild. La suppression d'une balise supprime uniquement son association au groupe de rapports, mais pas la balise en elle-même.

Note

Si vous supprimez un groupe de rapports CodeBuild, toutes les associations de balises sont supprimés du groupe de rapports supprimé. Vous n'avez pas besoin de supprimer les balises avant de supprimer un groupe de rapports.

Pour supprimer une ou plusieurs balises d'un groupe de rapports, reportez-vous à la section [Modifier les balises d'un groupe de rapports \(AWS CLI\)](#). Mettez à jour la tags section dans les données formatées JSON avec une liste mise à jour de balises qui ne contient pas celles que vous souhaitez supprimer. Si vous souhaitez supprimer toutes les balises, mettez à jour la section tags pour :

```
"tags: []"
```

Utilisation des groupes de rapports partagés.

Le partage de groupes de rapports permet à plusieurs comptes ou utilisateurs AWS d'afficher un groupe de rapports, ses rapports non expirés et les résultats des tests de ses rapports. Dans ce modèle, le compte propriétaire du groupe de rapports (propriétaire) partage un groupe de rapports avec d'autres comptes (consommateurs). Un consommateur ne peut pas modifier un groupe de rapports. Un rapport expire 30 jours après sa création.

Table des matières

- [Prérequis pour le partage de groupes de rapports](#)
- [Prérequis pour l'accès aux groupes de rapports partagés avec vous](#)
- [Services connexes](#)
- [Partage d'un groupe de rapports](#)
- [Annulation du partage d'un groupe de rapports partagé](#)
- [Identification d'un groupe de rapports partagé](#)
- [Autorisations de groupe de rapports partagés](#)

Prérequis pour le partage de groupes de rapports

Pour partager un groupe de rapports, votre compte AWS doit en être le propriétaire. Vous ne pouvez pas partager un groupe de rapports qui a été partagé avec vous.

Prérequis pour l'accès aux groupes de rapports partagés avec vous

Pour accéder à un groupe de rapports partagé, le rôle IAM d'un consommateur requiert l'autorisation `BatchGetReportGroups`. Vous pouvez attacher la stratégie suivante à leur rôle IAM :

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetReportGroups"
  ]
}
```

Pour plus d'informations, consultez [Utilisation de politiques basées sur l'identité pour AWS CodeBuild](#).

Services connexes

Le partage de groupe de rapports s'intègre à AWS Resource Access Manager (AWS RAM), un service qui vous permet de partager vos ressources AWS avec n'importe quel compte AWS ou via AWS Organizations. Avec AWS RAM, vous partagez les ressources que vous possédez en créant un partage de ressources qui spécifie les ressources et les consommateurs avec qui vous souhaitez les partager. Les consommateurs peuvent être des comptes individuels AWS, des

unités organisationnelles dans AWS Organizations ou l'ensemble d'une organisation dans AWS Organizations.

Pour plus d'informations, veuillez consulter le Guide de l'utilisateur [AWS RAM](#).

Partage d'un groupe de rapports

Lorsque vous partagez un groupe de rapports, le consommateur dispose d'un accès en lecture seule au groupe de rapports et à ses rapports. Le consommateur peut utiliser le AWS CLI pour afficher le groupe de rapports, ses rapports et les résultats de cas de test pour chaque rapport. Le consommateur ne peut pas :

- Afficher un groupe de rapports partagé ou ses rapports dans la console CodeBuild.
- Modifier un groupe de rapports partagé.
- Utiliser l'ARN du groupe de rapports partagé dans un projet pour exécuter un rapport. Une génération de projet qui spécifie un groupe de rapports partagé échoue.

Vous pouvez utiliser la console CodeBuild pour ajouter un groupe de rapports à un partage de ressources existant. Si vous souhaitez ajouter le groupe de rapports à un nouveau partage de ressources, vous devez d'abord le créer dans la [console AWS RAM](#).

Pour partager un groupe de rapports avec des unités organisationnelles ou une organisation entière, vous devez activer le partage avec AWS Organizations. Pour de plus amples informations, veuillez consulter [Activer le partage avec AWS Organizations](#) dans le Guide de l'utilisateur AWS RAM.

Vous pouvez utiliser la console CodeBuild, AWS RAM ou AWS CLI pour partager les groupes de rapports que vous possédez.

Pour partager un groupe de rapports que vous possédez (console CodeBuild)

1. Ouverture d'AWS CodeBuild Console à <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le projet que vous souhaitez partager, puis choisissez Partager. Pour de plus amples informations, veuillez consulter [Création d'un partage de ressources](#) dans le Guide de l'utilisateur AWS RAM.

Pour partager des groupes de rapports que vous possédez (console AWS RAM)

Voir [Créer un partage de ressources](#) dans le AWS RAM Guide de l'utilisateur.

Pour partager les groupes de rapports que vous possédez (commande AWS RAM)

Utilisez la commande [create-resource-share](#).

Pour partager un groupe de rapports que vous possédez (commande CodeBuild)

Utilisez la commande [put-resource-policy](#) :

1. Créez un fichier nommé `policy.json` et copiez ce qui suit dans celui-ci.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "consumer-aws-account-id-or-user"
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"
    ],
    "Resource": "arn-of-report-group-to-share"
  }]
}
```

2. Mettez à jour `policy.json` avec l'ARN du groupe de rapports et les identifiants avec lesquels vous le partagez. Dans l'exemple suivant, un accès en lecture seule est accordé au groupe de rapports avec l'ARN `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` à Alice et à l'utilisateur racine du AWS compte identifié par `123456789012`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::123456789012:user/Alice",
        "123456789012"
      ]
    }
  ]
}
```

```
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"],
    "Resource": "arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-
group"
  ]
}
```

3. Exécutez la commande suivante .

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://
policy.json
```

Annulation du partage d'un groupe de rapports partagé

Un groupe de rapports non partagé, y compris ses rapports et leurs résultats de cas de test, est accessible uniquement par son propriétaire. Si vous désactivez le partage d'un groupe de rapports, tout compte ou utilisateur AWS avec lequel vous l'avez précédemment partagé ne peut pas accéder au groupe de rapports, à ses rapports ou aux résultats des tests dans les rapports.

Pour annuler le partage d'un groupe de rapports partagé qui vous appartient, vous devez le supprimer du partage de ressources. Vous pouvez utiliser la console AWS RAM ou AWS CLI pour le faire.

Pour annuler le partage d'un groupe de rapports partagé que vous possédez (console AWS RAM)

Consultez la section [Mise à jour d'un partage de ressources](#) du Guide de l'utilisateur AWS RAM.

Pour annuler le partage d'un groupe de rapports partagé que vous possédez (commande AWS RAM)

Utilisez la commande [disassociate-resource-share](#).

Pour supprimer le partage d'un groupe de rapports dont vous possédez la commande CodeBuild)

Exécutez la commande [delete-resource-policy](#) et spécifiez l'ARN du groupe de rapports dont vous souhaitez annuler le partage :

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Identification d'un groupe de rapports partagé

Les propriétaires et les consommateurs peuvent utiliser le AWS CLI pour identifier les groupes de rapports partagés.

Pour identifier et obtenir des informations sur un groupe de rapports partagé et ses rapports, utilisez les commandes suivantes :

- Pour afficher les ARN des groupes de rapports partagés avec vous, exécutez [list-shared-report-groups](#) :

```
aws codebuild list-shared-report-groups
```

- Pour afficher les ARN des rapports dans un groupe de rapports, exécutez [list-reports-for-report-group](#) à l'aide de l'ARN du groupe de rapports :

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Pour afficher des informations sur les cas de test dans un rapport, exécutez [describe-test-cases](#) à l'aide de l'ARN de rapport :

```
aws codebuild describe-test-cases --report-arn report-arn
```

Le résultat se présente comme suit :

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

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

Autorisations de groupe de rapports partagés

Autorisations accordées aux propriétaires

Un propriétaire de groupe de rapports peut modifier le groupe de rapports et le spécifier dans un projet pour exécuter des rapports.

Autorisations accordées aux consommateurs

Un consommateur de groupe de rapports peut afficher un groupe de rapports, ses rapports et les résultats de cas de test pour ses rapports. Un consommateur ne peut pas modifier un groupe de rapports ou ses rapports, et ne peut pas l'utiliser pour créer des rapports.

Utilisation des rapports

Un rapport contient les résultats des cas de test spécifiés pour un groupe de rapports. Un rapport de test est créé lors de l'exécution d'un projet de génération. Vous spécifiez un groupe de rapports, des fichiers de cas de test et des commandes pour exécuter les cas de test dans son fichier buildspec. Chaque fois que les cas de test sont exécutés, un nouveau rapport de test est créé dans le groupe de rapports.

Un rapport de test expire 30 jours après sa création. Vous ne pouvez pas afficher un rapport de test expiré, mais vous pouvez exporter les résultats de test vers des fichiers de résultats de test bruts dans un compartiment S3. Les fichiers de test bruts exportés n'expirent pas. Pour de plus amples informations, veuillez consulter [Mise à jour d'un groupe de rapports](#).

L'état d'un rapport de test peut être l'un des suivants :

- **GENERATING** : l'exécution des cas de test est toujours en cours.

- **DELETING** : le rapport de test est en cours de suppression. Lorsqu'un rapport de test est supprimé, ses cas de test sont également supprimés. Les fichiers de données de résultats de test bruts exportés vers un compartiment S3 ne sont pas supprimés.
- **INCOMPLETE** : le rapport de test n'a pas été réalisé. Cet état peut être renvoyé pour l'une des raisons suivantes :
 - Problème de configuration du groupe de rapports qui spécifie les cas de test de ce rapport. Par exemple, le chemin d'accès aux cas de test sous le groupe de rapports dans le fichier buildspec peut être incorrect.
 - L'utilisateur IAM qui a exécuté la génération n'a pas les autorisations nécessaires pour exécuter des tests. Pour de plus amples informations, veuillez consulter [Utilisation des autorisations de rapport de test](#).
 - La génération n'a pas été terminée en raison d'une erreur qui n'est pas liée aux tests.
- **SUCCEEDED** : tous les cas de test ont été réussis.
- **FAILED** : certains des cas de test n'ont pas réussi.

Chaque cas de test renvoie un état. L'état d'un cas de test peut être l'un des suivants :

- **SUCCEEDED** : le cas de test a réussi.
- **FAILED** : le cas de test a échoué.
- **ERROR** : le cas de test a entraîné une erreur inattendue.
- **SKIPPED** : le cas de test ne s'est pas exécuté.
- **UNKNOWN** : le cas de test a renvoyé un état autre que SUCCEEDED, FAILED, ERROR ou SKIPPED.

Un rapport de test peut comporter un maximum de 500 résultats de cas de test. Si plus de 500 scénarios de test sont exécutés, CodeBuild hiérarchise les tests avec le statut FAILED et tronque les résultats du scénario de test.

Utilisation des autorisations de rapport de test

Cette rubrique décrit des informations importantes sur les autorisations liées aux rapports de test.

Rubriques

- [Création d'un rôle pour les rapports de test](#)
- [Autorisations pour les opérations de rapports de test](#)

- [Exemples d'autorisations de rapport de test](#)

Création d'un rôle pour les rapports de test

Pour exécuter un rapport de test et mettre à jour un projet de façon à inclure des rapports de test, votre rôle IAM requiert les autorisations suivantes. Ces autorisations sont incluses dans les politiques AWS gérées prédéfinies. Si vous souhaitez ajouter des rapports de test à un projet de génération existant, vous devez ajouter ces autorisations vous-même.

- `CreateReportGroup`
- `CreateReport`
- `UpdateReport`
- `BatchPutTestCases`

Pour exécuter un rapport de couverture de code, votre rôle IAM doit également inclure l'`BatchPutCodeCoverages` autorisation.

Note

`BatchPutTestCases`, `CreateReportUpdateReport`, et ne `BatchPutCodeCoverages` sont pas des autorisations publiques. Vous ne pouvez pas appeler une AWS CLI commande ou une méthode du SDK correspondante pour ces autorisations.

Pour vous assurer que vous disposez de ces autorisations, vous pouvez associer la politique suivante à votre rôle IAM :

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

```
]
}
```

Nous vous recommandons de limiter cette stratégie aux groupes de rapports que vous devez utiliser. Les éléments suivants restreignent les autorisations aux groupes de rapports contenant les deux ARN dans la stratégie :

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Ce qui suit restreint les autorisations aux groupes de rapports créés en exécutant des versions d'un projet nommé `my-project` :

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Note

Le rôle CodeBuild de service spécifié dans le projet est utilisé pour les autorisations de téléchargement vers le compartiment S3.

Autorisations pour les opérations de rapports de test

Vous pouvez spécifier des autorisations pour les opérations d' API CodeBuild de rapports de test suivantes :

- BatchGetReportGroups
- BatchGetReports
- CreateReportGroup
- DeleteReportGroup
- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

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

Exemples d'autorisations de rapport de test

Pour de plus amples informations sur les exemples de stratégies liées aux rapports de test, veuillez consulter les rubriques suivantes :

- [Autorisation d'un utilisateur à modifier un groupe de rapports](#)
- [Autorisation d'un utilisateur à créer un groupe de rapports](#)
- [Autorisation d'un utilisateur à supprimer un rapport](#)
- [Autorisation d'un utilisateur à supprimer un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports](#)

- [Autorisation d'un utilisateur à obtenir des informations sur les rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport](#)

Affichage des rapports de test

Vous pouvez afficher des détails sur un rapport de test, tels que des informations sur ses cas de test, ses nombres de réussite et d'échec, et la durée de son exécution. Vous pouvez consulter les rapports de test regroupés par build, par groupe de rapports ou par AWS compte. Choisissez un rapport de test dans la console pour afficher les détails et les résultats de ses scénarios de test.

Vous pouvez afficher les rapports de test qui n'ont pas expiré. Les rapports de test expirent 30 jours après leur création. Vous ne pouvez pas afficher un rapport expiré dans CodeBuild.

Rubriques

- [Affichage des rapports de test d'une génération](#)
- [Affichage des rapports de test d'un groupe de rapports](#)
- [Affichage des rapports de test dans votre compte AWS](#)

Affichage des rapports de test d'une génération

Pour afficher les rapports de test d'une génération

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Localisez la génération que vous souhaitez afficher. Si vous connaissez le projet qui a exécuté la génération qui a créé le rapport de test :
 1. Dans le panneau de navigation, choisissez Créer des projets, puis choisissez le projet avec la génération qui a exécuté le rapport de test que vous souhaitez afficher.
 2. Choisissez Générer l'historique, puis choisissez la génération exécutée qui a créé les rapports que vous souhaitez afficher.

Vous pouvez également localiser la génération dans l'historique de génération de votre compte AWS :

1. Dans le panneau de navigation, choisissez Historique de génération, puis choisissez la génération qui a créé les rapports de test que vous souhaitez afficher.
3. Dans la page de génération, choisissez Rapports, puis choisissez un rapport de test pour afficher ses détails.

Affichage des rapports de test d'un groupe de rapports

Pour afficher des rapports de test dans un groupe de rapports

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le panneau de navigation, sélectionnez Groupes de rapports.
3. Choisissez le groupe de rapports qui contient les rapports de test à afficher.
4. Choisissez un rapport de test pour voir ses détails.

Affichage des rapports de test dans votre compte AWS

Pour consulter les rapports de test dans votre AWS compte

1. Ouvrez la AWS CodeBuild console à l'[adresse https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Dans le panneau de navigation, sélectionnez Historique des événements.
3. Choisissez un rapport de test pour voir ses détails.

Rapport de test avec des cadres de test

Les rubriques de cette section montrent comment configurer les rapports de test dans AWS CodeBuild différents frameworks de test.

Rubriques

- [Configurer les rapports de test avec Jasmine](#)
- [Configurer les rapports de test avec Jest](#)
- [Configurer les rapports de test avec pytest](#)
- [Configurer les rapports de test avec RSpec](#)

Configurer les rapports de test avec Jasmine

La procédure suivante montre comment configurer les rapports de test dans AWS CodeBuild avec le [cadre de test JasmineBDD](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un projet CodeBuild existant.
- Votre projet est un projet Node.js qui est configuré pour utiliser le cadre de test Jasmine.

Ajoutez le package [jasmine-reporters](#) à la section `devDependencies` du fichier `package.json` de votre projet. Ce package contient une collection de classes de reporters JavaScript qui peuvent être utilisées avec Jasmine.

```
npm install --save-dev jasmine-reporters
```

S'il n'est pas déjà présent, ajoutez le script `test` au fichier `package.json` de votre projet. Le script garantit que Jasmine est appelé quand `npm test` est exécuté.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild prend en charge les reporters de test Jasmine suivants :

JUnitXmlReporter

Permet de générer des rapports au format `JUnitXml`.

NUnitXmlReporter

Permet de générer des rapports au format `NunitXml`.

Un projet Node.js avec Jasmine aura, par défaut, un sous-répertoire `spec`, qui contient la configuration de Jasmine et les scripts de test.

Pour configurer Jasmine pour générer des rapports au JUnitXML format, instanciez le `JUnitXmlReporter` rapporteur en ajoutant le code suivant à vos tests.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(junitReporter);
```

Pour configurer Jasmine pour générer des rapports au NUnitXML format, instanciez le `NUnitXmlReporter` rapporteur en ajoutant le code suivant à vos tests.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(nunitReporter)
```

Les rapports de test sont exportés dans le fichier spécifié par *<répertoire des rapports de test>/<nom de fichier des rapports>*.

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
```



```
commands:
  - npm build
  - npm test

reports:
  jasmine_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Si vous utilisez le format de rapport NunitXml, modifiez la valeur `file-format` comme suit.

```
file-format: NUNITXML
```

Configurer les rapports de test avec Jest

La procédure suivante montre comment configurer les rapports de test dans AWS CodeBuild avec le [cadre de test Jest](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un projet CodeBuild existant.
- Votre projet est un projet Node.js qui est configuré pour utiliser le cadre de test Jest.

Ajoutez le package [jest-junit](#) à la section `devDependencies` du fichier `package.json` de votre projet. CodeBuild utilise ce package pour générer des rapports dans le `JunitXml`.

```
npm install --save-dev jest-junit
```

S'il n'est pas déjà présent, ajoutez le script `test` au fichier `package.json` de votre projet. Le `test` garantit que Jest est appelé quand `npm test` est exécuté.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Configurez Jest pour utiliser le reporter JUnitXml en ajoutant ce qui suit à votre fichier de configuration Jest. Si votre projet ne possède pas de fichier de configuration Jest, créez un fichier nommé `jest.config.js` à la racine de votre projet et ajoutez ce qui suit. Les rapports de test sont exportés dans le fichier spécifié par *<répertoire des rapports de test>/<nom de fichier des rapports>*.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
  ]
};
```

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jest_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Configurer les rapports de test avec pytest

La procédure suivante montre comment configurer les rapports de test dans AWS CodeBuild avec le [cadre de test pytest](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un projet CodeBuild existant.
- Votre projet est un projet Python qui est configuré pour utiliser le cadre de test pytest.

Ajoutez l'entrée suivante à la phase `build` ou `post_build` de votre fichier `buildspec.yml`. Ce code découvre automatiquement les tests dans le répertoire courant et exporte les rapports de test dans le fichier spécifié par `<répertoire des rapports de test>/<nom de fichier des rapports>`. Le rapport utilise le format JunitXml.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

Dans votre fichier `buildspec.yml`, ajouter/mettez à jour les sections suivantes.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Configurer les rapports de test avec RSpec

La procédure suivante montre comment configurer les rapports de test dans AWS CodeBuild avec le [cadre de test RSpec](#).

La procédure nécessite les conditions préalables suivantes :

- Vous avez un projet CodeBuild existant.
- Votre projet est un projet Ruby qui est configuré pour utiliser le cadre de test RSpec.

Ajoutez/mettez à jour ce qui suit dans votre fichier `buildspec.yml`. Ce code exécute les tests dans le *<répertoire source des tests>* et exporte les rapports de test dans le fichier spécifié par *<répertoire des rapports de test>/<nom de fichier des rapports>*. Le rapport utilise le format JUnitXML.

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  pre_build:
    commands:
      - gem install rspec
      - gem install rspec_junit_formatter
  build:
    commands:
      - rspec <test source directory>/* --format RspecJUnitFormatter --out <test report
        <report filename>
  reports:
    rspec_reports:
      files:
        - <report filename>
      base-directory: <test report directory>
      file-format: JUNITXML
```

Rapports de couverture du code

CodeBuild vous permet de générer des rapports de couverture de code pour vos tests. Les rapports de couverture de code suivants sont fournis :

Couverture des lignes

La couverture de ligne mesure le nombre de relevés couverts par vos tests. Une déclaration est une instruction unique, sans inclure les commentaires ou les conditions.

$$\text{line coverage} = (\text{total lines covered}) / (\text{total number of lines})$$

Couverture des branches

La couverture des succursales mesure le nombre de branches que vos tests couvrent à partir de chaque branche possible d'une structure de contrôle, comme `unifoucase`.

```
branch coverage = (total branches covered)/(total number of branches)
```

Les formats de fichier de rapport de couverture du code suivants sont pris en charge :

- JaCoCo XML
- SimpleCov JSON¹
- Clover XML
- Cobertura XML

¹ CodeBuild accepte les rapports de couverture de code JSON générés par [simplecov](#), non [simplecov-json](#).

Création d'un rapport de couverture de code

Pour créer un rapport de couverture du code, vous exécutez un projet de construction qui est configuré avec au moins un groupe de rapports de couverture du code dans son fichier buildspec. CodeBuild interprétera les résultats de la couverture de code et fournira un rapport de couverture de code pour l'exécution. Un nouveau rapport de test est généré pour chaque génération ultérieure qui utilise le même fichier buildspec.

Pour créer un rapport de test

1. Créez un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération dans AWS CodeBuild](#).
2. Configurez le fichier buildspec de votre projet avec les informations du rapport de test :
 - a. Ajout d'un `reports` : et indiquez le nom de votre groupe de rapports. CodeBuild crée pour vous un groupe de rapports en utilisant le nom de votre projet et le nom que vous avez spécifié au format `project-name-report-group-name-in-buildspec`. Si vous avez déjà un groupe de rapports à utiliser, spécifiez son ARN. Si vous utilisez le nom au lieu de l'ARN, CodeBuild crée un groupe de rapports. Pour plus d'informations, consultez [Reports syntax in the buildspec file](#).
 - b. Sous le groupe de rapports, spécifiez l'emplacement des fichiers qui contiennent les résultats de couverture du code. Si vous utilisez plusieurs groupes de rapports, spécifiez les emplacements des fichiers de résultats pour chaque groupe de rapports. Un nouveau rapport de couverture du code est créé chaque fois que votre projet de génération s'exécute. Pour plus d'informations, consultez [Spécification des fichiers de test](#).

Il s'agit d'un exemple qui génère un rapport de couverture de code pour un fichier de résultats XML JaCoCo situé dans `test-results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

- c. Dans `commandsSection du buildou post_build`, spécifiez les commandes qui exécutent l'analyse de couverture de code. Pour plus d'informations, consultez [Spécification des commandes de test](#).
3. Exécutez une version du projet de génération. Pour plus d'informations, consultez [Exécution d'une génération dans AWS CodeBuild](#).
4. Une fois la génération terminée, choisissez la nouvelle version dans Historique de génération sur la page de votre projet. Choisissez `Rapports` pour afficher le rapport de couverture de code. Pour de plus amples informations, veuillez consulter [Affichage des rapports de test d'une génération](#).

Journalisation et surveillance dans AWS CodeBuild

La surveillance est un enjeu important pour assurer la fiabilité, la disponibilité et les performances d'AWS CodeBuild et de vos solutions AWS. Vous devez recueillir les données de surveillance de toutes les parties de votre AWS afin que vous puissiez déboguer plus facilement une éventuelle défaillance à plusieurs points. AWS fournit les outils suivants pour surveiller vos ressources et versions CodeBuild et répondre aux éventuels incidents.

Rubriques

- [Journalisation des appels d'API AWS CodeBuild avec AWS CloudTrail](#)
- [Surveillance des AWS CodeBuild](#)

Journalisation des appels d'API AWS CodeBuild avec AWS CloudTrail

AWS CodeBuild est intégré à AWS CloudTrail, service qui enregistre les actions effectuées par un utilisateur, un rôle ou un AWS service dans CodeBuild. CloudTrail capture tous les appels d'API CodeBuild en tant qu'événements, y compris les CodeBuild appels de code. CodeBuild Si vous créez un journal de suivi, vous pouvez activer la CloudTrail livraison continue des événements dans un compartiment CodeBuild. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Historique des événements. Les informations collectées par CloudTrail, vous permettent de déterminer quelle demande a été envoyée à CodeBuild, l'adresse IP source à partir de laquelle la demande a été effectuée, qui a effectué la demande, quand, ainsi que d'autres informations.

Pour en savoir plus CloudTrail, consultez le [Guide de AWS CloudTrail l'utilisateur](#).

AWS CodeBuild informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lors de la création de ce dernier. Quand une activité a lieu CodeBuild, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements AWS de service dans Event. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#) dans le Guide de AWS CloudTrail l'utilisateur.

Pour un enregistrement continu des événements dans votre AWS compte CodeBuild, créez un journal de suivi. Un journal CloudTrail de suivi permet de livrer des fichiers journaux dans un compartiment. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions. Le journal d'activité enregistre les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment S3 de votre choix. Vous pouvez configurer d'autres AWS services pour analyser plus en profondeur les données d'événement collectées CloudTrail dans les Pour plus d'informations, veuillez consulter les rubriques :

- [Présentation de la création d'un journal d'activité](#)
- [CloudTrail services et intégrations pris en charge](#)
- [Amazon SNS notifications CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions et Réception de fichiers journaux de plusieurs régions](#) et [Réception de fichiers CloudTrail journaux de plusieurs régions et Réception de fichiers](#)

Toutes les CodeBuild actions sont enregistrées CloudTrail et documentées dans la [référence deCodeBuild l'API](#). Par exemple, les appels aux actions `CreateProject` (dans leAWS CLI,`create-project`), `StartBuild` (dans leAWS CLI,`start-project`) et `UpdateProject` (dans leAWS CLI,`update-project`) génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez l'[élémentCloudTrail userIdentity](#) dans le Guide deAWS CloudTrail l'utilisateur.

Présentation des AWS CodeBuild entrées des fichiers journaux

Un journal de suivi est une configuration qui permet la remise d'événements sous forme de fichiers journaux dans un compartiment S3 que vous spécifiez. CloudTrail les fichiers journaux de suivi peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande

individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace de pile ordonnée des appels d'API.

Note

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

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

L'exemple suivant une entrée de CloudTrail journal qui illustre la création d'un projet de suivi 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"  
}
```

Surveillance des AWS CodeBuild

Vous pouvez utiliser Amazon CloudWatch pour surveiller vos générations de builds, signaler les problèmes et déclencher des actions automatiques, si nécessaire. Vous pouvez surveiller vos générations à deux niveaux :

Au niveau du projet

Ces métriques s'appliquent à toutes les générations du projet spécifié. Pour voir les métriques d'un projet, spécifiez `ProjectName` pour la dimension dans CloudWatch.

AWS niveau du compte

Ces métriques s'appliquent à toutes les générations d'un compte. Pour voir les mesures sur la page AWS au niveau du compte, n'indiquez pas de dimension dans CloudWatch. Les mesures d'utilisation des ressources de génération ne sont pas disponibles dans le AWS niveau du compte.

Les métriques CloudWatch montrent le comportement de vos générations au fil du temps. Par exemple, vous pouvez surveiller :

- Combien de générations ont été tentées dans un projet de génération ou un compte AWS au fil du temps.
- Combien de générations ont été réussies dans un projet de génération ou un compte AWS au fil du temps.
- Combien de générations ont échoué dans un projet de génération ou un compte AWS au fil du temps.
- Combien de temps CodeBuild a consacré à l'exécution de générations dans un projet de génération ou AWS compte au fil du temps.
- Générez l'utilisation des ressources pour une construction ou un projet de construction complet. Les métriques d'utilisation des ressources incluent des métriques telles que l'UC, la mémoire et l'utilisation du stockage.

Pour plus d'informations, consultez [Surveillance des métriques CodeBuild](#).

métriques CodeBuild CloudWatch

Les métriques suivantes peuvent être suivies par AWS compte ou projet de construction.

Durée de construction

Mesure la durée de la phase BUILD de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Versions

Mesure le nombre de générations déclenchées.

Unités: Nombre

Statistiques CloudWatch valides : Somme

Durée du téléchargement de la source

Mesure la durée de la phase DOWNLOAD_SOURCE de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée

Mesure la durée de toutes les générations au fil du temps.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

FailedBuilds

Mesure le nombre de générations qui ont échoué en raison d'une erreur client ou d'un délai d'expiration.

Unités: Nombre

Statistiques CloudWatch valides : Somme

Durée de la finalisation

Mesure la durée de la phase FINALIZING de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée d'installation

Mesure la durée de la phase INSTALL de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée de post-construction

Mesure la durée de la phase POST_BUILD de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée de pré-construction

Mesure la durée de la phase PRE_BUILD de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée du provisionnement

Mesure la durée de la phase PROVISIONING de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée de la file d'attente

Mesure la durée de la phase QUEUED de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Durée soumise

Mesure la durée de la phase SUBMITTED de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

SucceededBuilds

Mesure le nombre de générations réussies.

Unités: Nombre

Statistiques CloudWatch valides : Somme

Durée du téléchargement des artefacts

Mesure la durée de la phase UPLOAD_ARTIFACTS de la génération.

Unités: Secondes

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Mesures d'utilisation des ressources CodeBuild CloudWatch

Note

Les métriques d'utilisation des ressources CodeBuild sont uniquement disponibles dans les régions suivantes :

- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Asia Pacific (Mumbai)
- Région Asia Pacific (Singapour)
- Région Asia Pacific (Sydney)
- Région Canada (Centre)
- Région Europe (Frankfurt)
- Région Europe (Irlande)
- Région Europe (Londres)
- Région Europe (Paris)
- Région South America (São Paulo)
- US East (N. Virginia) Region
- Région US East (Ohio)

- Région US West (N. California)
- Région US West (Oregon)

Les mesures d'utilisation des ressources suivantes peuvent être suivies.

CPU utilisé

Nombre d'unités UC du traitement alloué utilisées par le conteneur de génération.

Unités: Unités CPU

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Pourcentage utilisé par le processeur

Pourcentage de traitement alloué utilisé par le conteneur de génération.

Unités: Pourcentage

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

MemoryUtilized

Nombre de mégaoctets de mémoire utilisés par le conteneur de génération.

Unités: Mégaoctets

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

Pourcentage de mémoire utilisée

Pourcentage de mémoire allouée utilisé par le conteneur de génération.

Unités: Pourcentage

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

StorageReadBytes

Vitesse de lecture du stockage utilisée par le conteneur de construction.

Unités: Octets/seconde

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

StorageWriteBytes

Vitesse d'écriture du stockage utilisée par le conteneur de génération.

Unités: Octets/seconde

Statistiques CloudWatch valides : Moyenne (recommandé), maximum, minimum

CodeBuild CloudWatch

CodeBuild fournit les dimensions de mesure CloudWatch suivantes. Si aucun de ces éléments n'est spécifié, les métriques s'appliquent auAWS.

BuildId, BuildNumber, ProjectName

Les mesures sont fournies pour un identifiant de build, un numéro de build et un nom de projet.

ProjectName

Les métriques sont fournies pour un nom de projet.

Des alarmes CodeBuild CloudWatch

Vous pouvez utiliser la console CloudWatch pour créer des alarmes basées sur les métriques CodeBuild, afin de pouvoir réagir en cas de problème avec vos générations. Les deux métriques les plus utiles en association avec des alarmes sont les suivantes :

- **FailedBuild.** Vous pouvez créer une alarme qui se déclenche quand un certain nombre de générations ayant échoué sont détectées au cours d'une période en secondes prédéterminée. Dans CloudWatch, vous spécifiez le nombre de secondes et le nombre d'échecs de génération nécessaires pour déclencher une alarme.
- **Duration.** Vous pouvez créer une alarme qui se déclenche quand une génération prend plus de temps que prévu. Vous spécifiez le nombre de secondes qui doit s'écouler entre le démarrage et la fin d'une génération, avant le déclenchement de l'alarme.

Pour plus d'informations sur la création d'alarmes pour les métriques CodeBuild, consultez [Surveillance des générations à l'aide des alarmes CloudWatch](#). Pour plus d'informations sur les alarmes, consultez [Création d'alarmes Amazon CloudWatch](#) dans le Guide de l'utilisateur Amazon CloudWatch..

Surveillance des métriques CodeBuild

AWS CodeBuild surveille les fonctions en votre nom et présente les métriques via Amazon CloudWatch. Ces métriques incluent le nombre total de générations, d'échecs et de réussites, ainsi que la durée des générations.

Pour surveiller des métriques concernant CodeBuild, vous pouvez utiliser la console CodeBuild ou la console CloudWatch. Les procédures suivantes vous expliquent comment accéder aux métriques.

Rubriques

- [Accès aux métriques de génération \(console CodeBuild\)](#)
- [Accès aux métriques de génération \(console Amazon CloudWatch\)](#)

Accès aux métriques de génération (console CodeBuild)

Note

Vous ne pouvez pas personnaliser les métriques ni les graphiques utilisés pour les afficher dans la console CodeBuild. Pour personnaliser l'affichage, utilisez la console Amazon CloudWatch pour visualiser vos métriques de génération.

Métriques au niveau du compte

Pour accéder à AWS Métriques au niveau du compte

1. Connectez-vous à la console AWS Management Console et ouvrez AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, sélectionnez Métriques de compte.

Métriques au niveau du projet

Pour accéder aux métriques au niveau du projet

1. Connectez-vous à la console AWS Management Console et ouvrez AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.

3. Dans la liste des projets de génération, dans la colonne Nom, choisissez le projet dont vous souhaitez consulter les métriques.
4. Choisissez l'onglet Métriques.

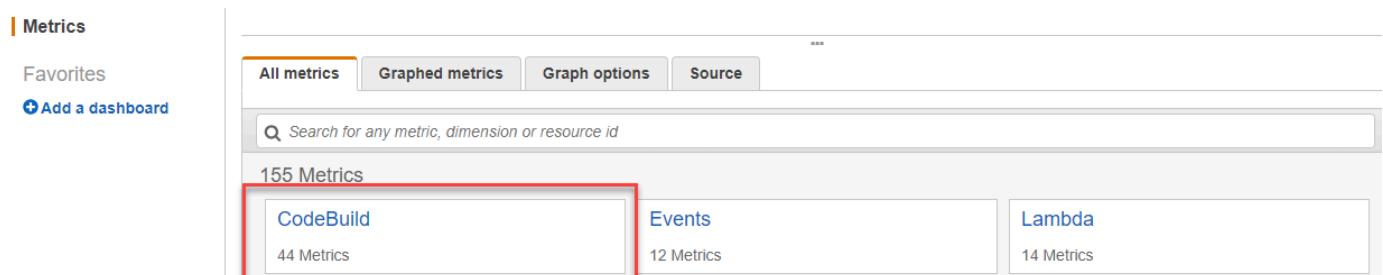
Accès aux métriques de génération (console Amazon CloudWatch)

Vous pouvez personnaliser les métriques et les graphiques utilisés pour les afficher avec la console CloudWatch.

Métriques au niveau du compte

Pour accéder aux métriques au niveau du compte

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Sous l'onglet Toutes les métriques, choisissez CodeBuild.

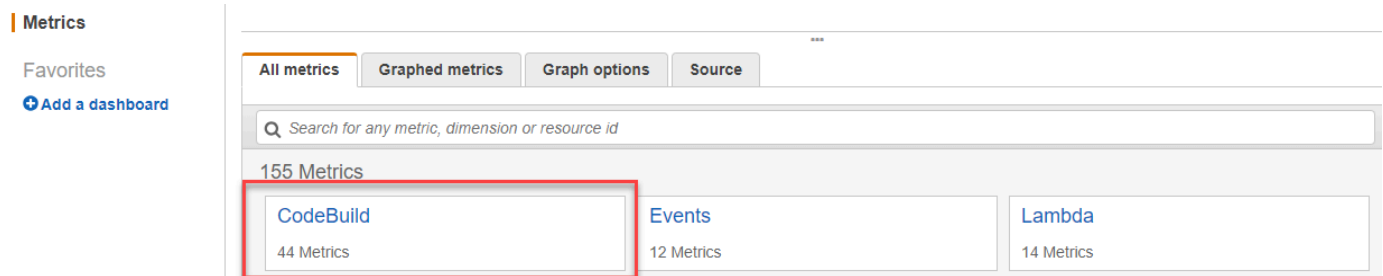


4. Choisissez Account metrics (Métriques globales).
5. Choisissez un ou plusieurs projets et métriques. Pour chaque projet, vous pouvez sélectionner les métriques SucceededBuilds, FailedBuilds, Builds et Duration. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.

Métriques au niveau du projet

Pour accéder aux métriques au niveau du projet

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Sous l'onglet Toutes les métriques, choisissez CodeBuild.



4. Choisissez Par projet.
5. Choisissez une ou plusieurs combinaisons de projets et de métriques. Pour chaque projet, vous pouvez sélectionner les métriques SucceededBuilds, FailedBuilds, Builds et Duration. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.
6. (Facultatif) Vous pouvez personnaliser vos métriques et graphiques. Par exemple, dans la liste déroulante du Statistique, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour de plus amples informations, veuillez consulter [Graphique de métriques](#) et [Affichage des métriques disponibles](#) dans le Guide de l'utilisateur Amazon CloudWatch..


Surveillance des mesures d'utilisation des ressources CodeBuild

AWS CodeBuild surveille l'utilisation des ressources de génération en votre nom et présente les métriques via Amazon CloudWatch. Il s'agit notamment des métriques telles que l'UC, la mémoire et l'utilisation du stockage.

Note

Les mesures d'utilisation des ressources CodeBuild sont enregistrées uniquement pour les versions qui s'exécutent pendant plus d'une minute.

Vous pouvez utiliser la console CodeBuild ou la console CloudWatch pour surveiller les mesures d'utilisation des ressources pour CodeBuild.

 Note

Les métriques d'utilisation des ressources CodeBuild sont uniquement disponibles dans les régions suivantes :

- Région Asia Pacific (Tokyo)
- Région Asia Pacific (Seoul)
- Région Asia Pacific (Mumbai)
- Région Asia Pacific (Singapore)
- Région Asia Pacific (Sydney)
- Région Canada (Centre)
- Région Europe (Frankfurt)
- Région Europe (Ireland)
- Région Europe (Londres)
- Région Europe (Paris)
- Région South America (São Paulo)
- US East (N. Virginia) Region
- Région US East (Ohio)
- Région US West (N. California)
- Région US West (Oregon)

Les procédures suivantes vous expliquent comment accéder aux métriques d'utilisation des ressources.

Rubriques

- [Accès aux mesures d'utilisation des ressources \(console CodeBuild\)](#)
- [Accès aux mesures d'utilisation des ressources \(console Amazon CloudWatch\)](#)

Accès aux mesures d'utilisation des ressources (console CodeBuild)

Note

Vous ne pouvez pas personnaliser les métriques ni les graphiques utilisés pour les afficher dans la console CodeBuild. Pour personnaliser l'affichage, utilisez la console Amazon CloudWatch pour visualiser vos métriques de génération.

Métriques d'utilisation des ressources au niveau du projet

Pour accéder aux métriques d'utilisation des ressources au niveau du projet

1. Connectez-vous à la console AWS Management Console et ouvrez AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Projets de génération.
3. Dans la liste des projets de génération, dans la Nom, choisissez le projet pour lequel vous souhaitez afficher les métriques d'utilisation.
4. Choisissez l'onglet Métriques. Les mesures d'utilisation des ressources sont affichées dans le Métriques d'utilisation des ressources Section.
5. Pour afficher les métriques d'utilisation des ressources au niveau du projet dans la console CloudWatch, choisissez Afficher dans CloudWatch dans le Métriques d'utilisation des ressources Section.

Métriques d'utilisation des ressources au niveau du génération

Pour accéder aux métriques d'utilisation des ressources au niveau du génération

1. Connectez-vous à la console AWS Management Console et ouvrez AWS CodeBuild console <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dans le volet de navigation, choisissez Historique de génération.
3. Dans la liste des builds, dans le Exécution de génération, choisissez la génération pour laquelle vous souhaitez afficher les métriques d'utilisation.
4. Cliquez sur l'onglet Utilisation des ressources onglet

5. Pour afficher les métriques d'utilisation des ressources au niveau de la génération dans la console CloudWatch, choisissez Afficher dans CloudWatch dans le Métriques d'utilisation des ressources Section.

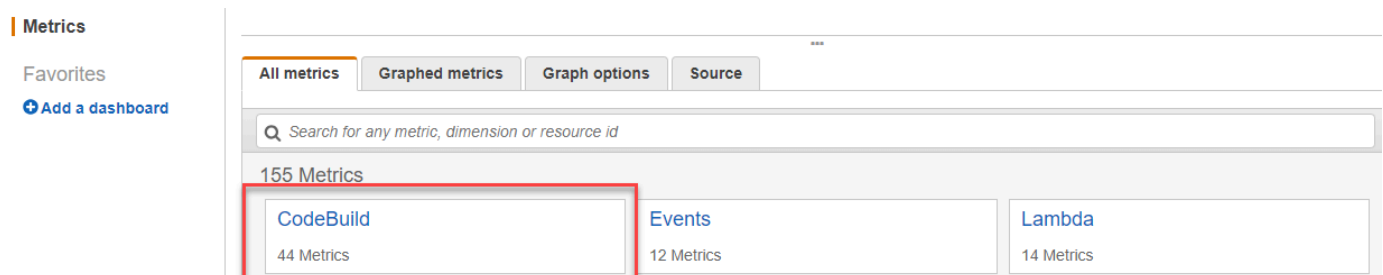
Accès aux mesures d'utilisation des ressources (console Amazon CloudWatch)

La console Amazon CloudWatch peut être utilisée pour accéder aux mesures d'utilisation des ressources CodeBuild.

Métriques d'utilisation des ressources au niveau du projet

Pour accéder aux métriques d'utilisation des ressources au niveau du projet

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Sous l'onglet Toutes les métriques, choisissez CodeBuild.



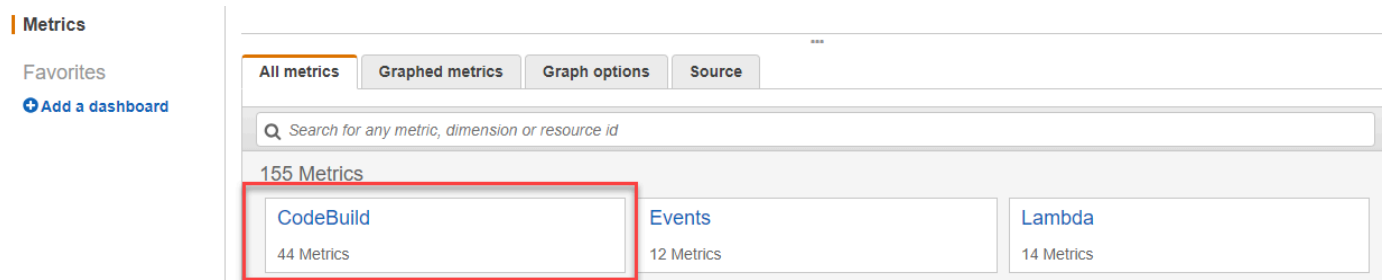
4. Choisissez Par projet.
5. Choisissez une ou plusieurs combinaisons de projets et de métriques à ajouter au graphique. Toutes les combinaisons de métriques et de projets sélectionnées sont affichées dans le graphique sur la page.
6. (Facultatif) Vous pouvez personnaliser vos métriques et graphiques à partir du Métriques graphiques onglet Par exemple, dans la liste déroulante du Statistique, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour de plus amples informations, veuillez consulter [Graphique des métriques](#) et [Affichage des métriques disponibles](#) dans le Guide de l'utilisateur Amazon CloudWatch..

Métriques d'utilisation des ressources au niveau du génération

Pour accéder aux métriques d'utilisation des ressources au niveau du génération

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Sous l'onglet Toutes les métriques, choisissez CodeBuild.



4. Choisissez BuildId Id, BuildNumber, NomProjectName.
5. Choisissez une ou plusieurs combinaisons de génération et de métriques à ajouter au graphique. Toutes les combinaisons de métriques et de génération sélectionnées sont affichées dans le graphique sur la page.
6. (Facultatif) Vous pouvez personnaliser vos métriques et graphiques à partir du Métriques graphiques onglet. Par exemple, dans la liste déroulante du Statistique, vous pouvez choisir une autre statistique à afficher. Dans le menu déroulant de la colonne Période, vous pouvez choisir une autre période à utiliser pour surveiller les métriques.

Pour de plus amples informations, veuillez consulter [Graphique des métriques](#) et [Affichage des métriques disponibles](#) dans le Guide de l'utilisateur Amazon CloudWatch..

Surveillance des générations à l'aide des alarmes CloudWatch

Vous pouvez créer une alarme CloudWatch pour vos générations. Une alarme surveille une métrique individuelle pendant une période que vous définissez et exécute une ou plusieurs actions en fonction de la valeur de cette métrique, par rapport à un seuil spécifié sur un certain nombre de périodes. En utilisant la fonctionnalité native d'alarme CloudWatch, vous pouvez spécifier n'importe quelle action prise en charge par CloudWatch lorsqu'un seuil est dépassé. Par exemple, vous pouvez demander qu'une notification Amazon SNS soit envoyée lorsque plus de trois générations dans votre compte échouent sur une période de 15 minutes.

Pour créer une alarme CloudWatch, pour une métrique CodeBuild

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le panneau de navigation, cliquez sur Alarms (Alarmes).
3. Sélectionnez Create Alarm (Créer une alarme).
4. Sous Métriques CloudWatch par catégorie, choisissez Métriques CodeBuild. Si vous savez que vous utiliserez uniquement les métriques de niveau projet, choisissez Par projet. Si vous savez que vous utiliserez uniquement les métriques de niveau compte, choisissez Métriques de compte.
5. Sous Créer une alarme, si l'option n'est pas déjà sélectionnée, choisissez Sélectionner une métrique.
6. Choisissez la métrique pour laquelle vous voulez créer une alarme. Les options disponibles sont Par projet et Métriques de compte.
7. Choisissez Suivant ou Définir une alarme, puis créez votre alarme. Pour plus d'informations, consultez [Création d'alarmes Amazon CloudWatch](#) dans le Guide de l'utilisateur Amazon CloudWatch. Pour plus d'informations sur la configuration des notifications Amazon SNS lorsqu'une alarme se déclenche, consultez [Configurer des notifications Amazon SNS](#) dans le Manuel du développeur Amazon SNS.
8. Sélectionnez Create Alarm (Créer une alarme).

Sécurité dans AWS CodeBuild

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité et la conformité sont une responsabilité partagée entre vous AWS et vous. Ce modèle partagé peut vous aider à alléger votre charge opérationnelle : il AWS exploite, gère et contrôle les composants depuis le système d'exploitation hôte et la couche de virtualisation jusqu'à la sécurité physique des installations de service. Vous assumez la responsabilité et la gestion du système d'exploitation invité (notamment les mises à jour et les correctifs de sécurité), d'autres logiciels d'application connexes. Vous êtes également responsable de la configuration du pare-feu du groupe de sécurité AWS fourni. Vos responsabilités dépendent des services que vous utilisez, de l'intégration de ces services dans votre environnement informatique et des lois et règlements applicables. Par conséquent, vous devez examiner soigneusement les services que votre organisation utilise. Pour de plus amples informations, veuillez consulter [Modèle de responsabilité partagée](#).

Pour savoir comment sécuriser vos CodeBuild ressources, consultez les rubriques suivantes.

Rubriques

- [Protection des données dans AWS CodeBuild](#)
- [Gestion des identités et des accès dans AWS CodeBuild](#)
- [Validation de conformité pour AWS CodeBuild](#)
- [Résilience dans AWS CodeBuild](#)
- [Sécurité de l'infrastructure dans AWS CodeBuild](#)
- [Accédez à votre fournisseur source dans CodeBuild](#)
- [Prévention du cas de figure de l'adjoint désorienté entre services](#)

Protection des données dans AWS CodeBuild

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS CodeBuild. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité

des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec CodeBuild ou d'autres Services AWS utilisateurs de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

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

- Chaînes spécifiées à l'aide du Parameter Store dans les variables d'environnement CodeBuild du projet ou dans la section `buildspecenv/parameter-store`. Pour plus d'informations, consultez la [présentation de la console Systems Manager Parameter Store et Systems Manager Parameter Store dans le guide](#) de l'utilisateur d'Amazon EC2 Systems Manager.
- Chaînes spécifiées AWS Secrets Manager à l'aide des variables d'environnement CodeBuild du projet ou de la section `buildspecenv/secrets-manager`. Pour de plus amples informations, veuillez consulter [Gestion des clés](#).

Pour de plus amples informations sur la protection des données, veuillez consulter le billet de blog [Modèle de responsabilité partagée AWS et RGPD](#) à la page Blog sur la sécurité AWS .

Rubriques

- [Chiffrement des données](#)
- [Gestion des clés](#)
- [Confidentialité du trafic](#)

Chiffrement des données

Le chiffrement est un élément important de la CodeBuild sécurité. Certains chiffrements, par exemple pour les données en transit, sont fournis par défaut et vous n'avez pas besoin de faire quoi que ce soit. Vous pouvez en configurer d'autres, par exemple pour les données au repos, lorsque vous créez votre projet ou votre génération.

- Chiffrement des données au repos - Les artefacts de build, tels qu'un cache, des journaux, des fichiers de données de rapports de test bruts exportés et des résultats de build, sont chiffrés par défaut à l'aide Clés gérées par AWS de. Si vous ne souhaitez pas utiliser ces clés KMS, vous devez créer et configurer une clé gérée par le client. Pour de plus amples informations, veuillez consulter [Création de clés KMS](#) et [Concepts d'AWS Key Management Service](#) dans le Guide de l'utilisateur AWS Key Management Service .
- Vous pouvez stocker l'identifiant de la clé AWS KMS CodeBuild utilisée pour chiffrer l'artefact de sortie de construction dans la variable d'`CODEBUILD_KMS_KEY_ID` environnement. Pour plus d'informations, consultez [Variables d'environnement dans les environnements de génération](#).
- Vous pouvez spécifier une clé gérée par le client lorsque vous créez un projet de construction. Pour plus d'informations, consultez [Set the Encryption Key Using the Console](#) et [Définissez la clé de chiffrement à l'aide de la CLI](#).

Les volumes Amazon Elastic Block Store de votre parc de produits sont chiffrés par défaut à l'aide de Clés gérées par AWS.

- Chiffrement des données en transit : toutes les communications entre les clients CodeBuild et entre CodeBuild et leurs dépendances en aval sont protégées par des connexions TLS signées selon le processus de signature Signature version 4. Tous les CodeBuild points de terminaison utilisent des certificats SHA-256 gérés par AWS Private Certificate Authority. Pour de plus amples informations, veuillez consulter [Processus de signature Signature Version 4](#) et [Présentation d'ACM PCA](#).
- Chiffrement des artefacts de génération : le rôle de CodeBuild service associé au projet de génération nécessite l'accès à une clé KMS afin de chiffrer ses artefacts de sortie de génération. Par défaut, CodeBuild utilise une Clé gérée par AWS pour Amazon S3 dans votre AWS compte. Si vous ne souhaitez pas l'utiliser Clé gérée par AWS, vous devez créer et configurer une clé gérée par le client. Pour plus d'informations, reportez-vous à [Création d'une clé gérée par le client](#) la section [Création de clés](#) dans le Guide du AWS KMS développeur.

Gestion des clés

Vous pouvez protéger votre contenu d'une utilisation non autorisée grâce au chiffrement. Stockez vos clés de chiffrement dans AWS Secrets Manager, puis autorisez le rôle de CodeBuild service associé au projet de génération à obtenir les clés de chiffrement depuis votre compte Secrets Manager.

Pour de plus amples informations, veuillez consulter [Création et configuration d'une clé gérée par le client pour CodeBuild](#), [Création d'un projet de génération dans AWS CodeBuild](#), [Exécution d'une génération dans AWS CodeBuild](#) et [Didacticiel : Stockage et récupération d'un secret](#).

Utilisez la variable d'environnement `CODEBUILD_KMS_KEY_ID` dans une commande de construction pour obtenir l'identifiant de AWS KMS clé. Pour de plus amples informations, veuillez consulter [Variables d'environnement dans les environnements de génération](#).

Vous pouvez utiliser Secrets Manager pour protéger les informations d'identification d'un registre privé qui stocke une image Docker utilisée pour votre environnement d'exécution. Pour de plus amples informations, veuillez consulter [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#).

Confidentialité du trafic

Vous pouvez améliorer la sécurité de vos builds en les configurant CodeBuild pour utiliser un point de terminaison VPC d'interface. Pour ce faire, vous n'avez pas besoin d'une passerelle Internet, d'un

périphérique NAT ni d'une passerelle privée virtuelle. Il n'est pas non plus nécessaire de le configurer PrivateLink, bien que cela soit recommandé. Pour de plus amples informations, veuillez consulter [Utilisation de points de terminaison de VPC](#). Pour plus d'informations sur PrivateLink les points de terminaison VPC, consultez [AWS PrivateLink](#) et [Accès aux AWS](#) services via. PrivateLink

Gestion des identités et des accès dans AWS CodeBuild

L'accès à AWS CodeBuild nécessite des informations d'identification. Ces informations d'identification doivent être autorisées à accéder aux AWS ressources, telles que le stockage et la récupération d'artefacts de build dans des compartiments S3 et la consultation d'Amazon CloudWatch Logs pour les builds. Les sections suivantes décrivent comment vous pouvez utiliser [AWS Identity and Access Management](#)(IAM) et comment CodeBuild sécuriser l'accès à vos ressources :

Vue d'ensemble de la gestion des autorisations d'accès à vos AWS CodeBuild ressources

Chaque AWS ressource appartient à un AWS compte, et les autorisations de création ou d'accès à une ressource sont régies par des politiques d'autorisation. Un compte administrateur peut attacher des politiques d'autorisations à des identités IAM (c'est-à-dire des utilisateurs, des groupes et des rôles).

Note

Un administrateur de compte (ou utilisateur administrateur) est un utilisateur doté des privilèges d'administrateur. Pour plus d'informations, consultez [Bonnes pratiques IAM](#) dans le Guide de l'utilisateur IAM.

Lorsque vous accordez des autorisations, vous décidez qui obtient les autorisations, les ressources auxquelles elles peuvent accéder et les actions pouvant être effectuées sur ces ressources.

Rubriques

- [AWS CodeBuild ressources et opérations](#)
- [Présentation de la propriété des ressources](#)
- [Gestion de l'accès aux ressources](#)
- [Spécification des éléments d'une politique : actions, effets et mandataires](#)

AWS CodeBuild ressources et opérations

Dans AWS CodeBuild, la ressource principale est un projet de construction. Dans une stratégie, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la stratégie s'applique. Les générations sont également des ressources et disposent d'ARN associés. Pour plus d'informations, consultez [Amazon Resource Names \(ARN\) et AWS Service Namespaces](#) dans le Référence générale d'Amazon Web Services.

| Type de ressource | Format ARN |
|--|--|
| Projet de génération | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :project/ <i>project-name</i> |
| Génération | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :build/ <i>build-ID</i> |
| Groupe de rapports | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report-group/ <i>report-group-name</i> |
| Rapport | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :report/ <i>report-ID</i> |
| Toutes les CodeBuild ressources | arn:aws:codebuild:* |
| Toutes les CodeBuild ressources détenues par le compte spécifié dans la AWS région spécifiée | arn:aws:codebuild: <i>region-ID</i> : <i>account-ID</i> :* |

Note

La plupart AWS des services traitent deux points (:)) ou une barre oblique (/) comme le même caractère dans les ARN. Cependant, CodeBuild utilise une correspondance exacte dans les modèles de ressources et les règles. Veillez à utiliser les caractères corrects lors

de la création de modèles d'événements, afin qu'ils correspondent à la syntaxe ARN de la ressource.

Par exemple, vous pouvez indiquer un projet de construction spécifique (*myBuildProject*) dans votre instruction à l'aide de son ARN comme suit :

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Pour spécifier toutes les ressources, ou si une action d'API ne prend pas en charge les ARN, utilisez le caractère générique (*) dans l'élément Resource comme suit :

```
"Resource": "*"
```

Certaines actions CodeBuild d'API acceptent plusieurs ressources (par exemple, `BatchGetProjects`). Pour spécifier plusieurs ressources dans une seule déclaration, séparez leurs ARN par des virgules comme suit :

```
"Resource": [  
  "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",  
  "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"  
]
```

CodeBuild fournit un ensemble d'opérations permettant de travailler avec les CodeBuild ressources. Pour en obtenir la liste, veuillez consulter [AWS CodeBuild référence aux autorisations](#).

Présentation de la propriété des ressources

Le AWS compte possède les ressources créées dans le compte, quelle que soit la personne qui les a créées. Plus précisément, le propriétaire de la ressource est le AWS compte de l'[entité principale](#) (c'est-à-dire le compte root, un utilisateur ou un rôle IAM) qui authentifie la demande de création de ressource. Les exemples suivants illustrent comment cela fonctionne :

- Si vous utilisez les informations d'identification du compte root de votre AWS compte pour créer une règle, votre AWS compte est le propriétaire de la CodeBuild ressource.
- Si vous créez un utilisateur dans votre AWS compte et que vous accordez l'autorisation de créer CodeBuild des ressources à cet utilisateur, celui-ci peut créer CodeBuild des ressources.

Cependant, votre AWS compte, auquel appartient l'utilisateur, est propriétaire des CodeBuild ressources.

- Si vous créez un rôle IAM dans votre AWS compte avec les autorisations nécessaires pour créer CodeBuild des ressources, toute personne pouvant assumer ce rôle peut créer des CodeBuild ressources. Votre AWS compte, auquel appartient le rôle, est propriétaire des CodeBuild ressources.

Gestion de l'accès aux ressources

Une politique d'autorisations décrit qui a accès à quelles ressources.

Note

Cette section décrit l'utilisation d'IAM dans AWS CodeBuild. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, consultez [Qu'est-ce que IAM ?](#) dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des stratégies IAM, consultez [Référence de stratégie AWS IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques attachées à une identité IAM sont appelées des politiques basées sur l'identité (politiques IAM). Les politiques associées à une ressource sont appelées politiques basées sur les ressources. CodeBuild prend en charge les politiques basées sur l'identité et les politiques basées sur les ressources pour certaines API en lecture seule à des fins de partage de ressources entre comptes.

Accès sécurisé aux compartiments S3

Nous vous recommandons vivement d'inclure les autorisations suivantes dans votre rôle IAM afin de vérifier que le compartiment S3 associé à votre CodeBuild projet est votre propriété ou celle d'une personne de confiance. Ces autorisations ne sont pas incluses dans les politiques et les rôles AWS gérés. Vous devez les ajouter vous-même.

- `s3:GetBucketAc1`
- `s3:GetBucketLocation`

Si le propriétaire d'un compartiment S3 utilisé par votre projet change, vous devez vérifier que vous êtes toujours propriétaire du compartiment et mettre à jour les autorisations dans votre rôle IAM dans

le cas contraire. Pour plus d'informations, consultez [Ajouter des autorisations d' CodeBuild accès à un groupe ou à un utilisateur IAM](#) et [Création d'un rôle CodeBuild de service](#).

Spécification des éléments d'une politique : actions, effets et mandataires

Pour chaque AWS CodeBuild ressource, le service définit un ensemble d'opérations d'API. Pour accorder des autorisations pour ces opérations d'API CodeBuild , définissez un ensemble d'actions que vous pouvez spécifier dans une politique. Certaines opérations d'API peuvent exiger des autorisations pour plusieurs actions afin de réaliser l'opération d'API. Pour plus d'informations, consultez [AWS CodeBuild ressources et opérations](#) et [AWS CodeBuild référence aux autorisations](#).

Voici les éléments de base d'une politique :

- Ressource : vous utilisez un nom Amazon Resource Name (ARN) pour identifier la ressource à laquelle s'applique la politique.
- Action : vous utilisez des mots clés d'action pour identifier les opérations sur les ressources que vous souhaitez autoriser ou refuser. Par exemple, l'autorisation `codebuild:CreateProject` autorise l'utilisateur à effectuer l'opération `CreateProject`.
- Effet : vous spécifiez l'effet, qu'il s'agisse d'autoriser ou de refuser, lorsque l'utilisateur demande l'action. Si vous n'accordez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez également explicitement refuser l'accès à une ressource. Vous pouvez le faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une stratégie différente accorde cet accès.
- Principal — Dans les politiques basées sur l'identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur une ressource, vous spécifiez l'utilisateur, le compte, le service ou une autre entité qui doit recevoir les autorisations.

Pour en savoir plus sur la syntaxe des stratégies IAM et pour obtenir des descriptions, consultez [Référence de stratégie IAM AWS](#) dans le Guide de l'utilisateur IAM.

Pour un tableau présentant toutes les actions d' CodeBuild API et les ressources auxquelles elles s'appliquent, consultez le [AWS CodeBuild référence aux autorisations](#).

Utilisation de politiques basées sur l'identité pour AWS CodeBuild

Cette rubrique fournit des exemples de stratégies basées sur l'identité qui montrent comment un administrateur de compte peut lier des stratégies d'autorisations à des identités IAM (autrement dit,

des utilisateurs, des groupes et des rôles) et accorder ainsi des autorisations pour effectuer des opérations sur les ressources AWS CodeBuild .

Important

Nous vous recommandons de consulter d'abord les rubriques d'introduction qui expliquent les concepts de base et les options disponibles pour gérer l'accès à vos CodeBuild ressources. Pour de plus amples informations, veuillez consulter [Vue d'ensemble de la gestion des autorisations d'accès à vos AWS CodeBuild ressources](#).

Rubriques

- [Autorisations requises pour utiliser la console AWS CodeBuild](#)
- [Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry](#)
- [Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources](#)
- [AWS politiques gérées \(prédéfinies\) pour AWS CodeBuild](#)
- [CodeBuild politiques et notifications gérées](#)
- [CodeBuild mises à jour des politiques AWS gérées](#)
- [Exemples de politiques gérées par le client](#)

Voici un exemple de stratégie d'autorisations qui autorise un utilisateur à obtenir des informations sur des projets de génération uniquement dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Autorisations requises pour utiliser la console AWS CodeBuild

Un utilisateur qui utilise la AWS CodeBuild console doit disposer d'un ensemble minimal d'autorisations lui permettant de décrire les autres AWS ressources du AWS compte. Vous devez détenir les autorisations des services suivants :

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (si vous stockez votre code source dans un AWS CodeCommit dépôt)
- Amazon Elastic Container Registry (Amazon ECR) (si vous utilisez un environnement de construction qui repose sur une image Docker dans un référentiel Amazon ECR)

Note

Depuis le 26 juillet 2022, la politique IAM par défaut a été mise à jour. Pour de plus amples informations, veuillez consulter [Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (si vous utilisez un environnement de construction qui repose sur une image Docker dans un référentiel Amazon ECR)
- AWS Identity and Access Management (JE SUIS)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu.

Autorisations requises pour se connecter AWS CodeBuild à Amazon Elastic Container Registry

Depuis le 26 juillet 2022, AWS CodeBuild a mis à jour sa politique IAM par défaut pour les autorisations Amazon ECR. Les autorisations suivantes ont été supprimées de la politique par défaut :

```
"ecr:PutImage",  
"ecr:InitiateLayerUpload",  
"ecr:UploadLayerPart",
```

```
"ecr:CompleteLayerUpload"
```

Pour les CodeBuild projets créés avant le 26 juillet 2022, nous vous recommandons de mettre à jour votre politique avec la politique Amazon ECR suivante :

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

Pour plus d'informations sur la mise à jour de votre politique, consultez [Ajouter des autorisations d'CodeBuild accès à un groupe ou à un utilisateur IAM](#).

Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources

La AWS CodeBuild console utilise les actions d'API suivantes pour se connecter aux fournisseurs de sources (par exemple, les GitHub référentiels).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Vous pouvez associer des fournisseurs de sources (tels que GitHub des référentiels) à vos projets de génération à l'aide de la AWS CodeBuild console. Pour ce faire, vous devez d'abord ajouter les actions d'API précédentes aux politiques d'accès IAM associées à l'utilisateur que vous utilisez pour accéder à la AWS CodeBuild console.

Les actions d'API `ListConnectedOAuthAccounts`, `ListRepositories` et `PersistOAuthToken` ne sont pas conçues pour être appelées par votre code. Par conséquent, ces actions d'API ne sont pas incluses dans les AWS SDK AWS CLI et.

AWS politiques gérées (prédéfinies) pour AWS CodeBuild

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par AWS. Ces politiques AWS gérées accordent les autorisations nécessaires pour les cas d'utilisation courants afin que vous n'ayez pas à rechercher les autorisations

nécessaires. Les politiques gérées fournissent CodeBuild également des autorisations pour effectuer des opérations dans d'autres services, tels que IAM AWS CodeCommit, Amazon EC2, Amazon ECR, Amazon SNS et CloudWatch Amazon Events, conformément aux responsabilités des utilisateurs auxquels la politique en question a été accordée. Par exemple, il s'agit d'une `AWSCodeBuildAdminAccess` politique utilisateur de niveau administratif qui permet aux utilisateurs dotés de cette politique de créer et de gérer des règles relatives aux CloudWatch événements pour les versions de projets et des rubriques Amazon SNS pour les notifications relatives aux événements liés au projet (sujets dont les noms sont préfixés par `arn:aws:codebuild:`), ainsi que d'administrer des projets et des groupes de rapports dans. CodeBuild Pour plus d'informations, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à AWS CodeBuild.

`AWSCodeBuildAdminAccess`

Fournit un accès complet CodeBuild , y compris les autorisations pour administrer les projets de CodeBuild construction.

`AWSCodeBuildDeveloperAccess`

Permet d'accéder à l'administration du projet de build, CodeBuild mais ne l'autorise pas.

`AWSCodeBuildReadOnlyAccess`

Fournit un accès en lecture seule à. CodeBuild

Pour accéder aux artefacts de sortie de build CodeBuild créés, vous devez également joindre la politique AWS gérée nommée `AmazonS3ReadOnlyAccess`.

Pour créer et gérer des rôles de CodeBuild service, vous devez également associer la politique AWS gérée nommée `IAMFullAccess`.

Vous pouvez également créer vos propres politiques IAM personnalisées afin d'accorder des autorisations pour les actions et les ressources CodeBuild. Vous pouvez attacher ces stratégies personnalisées aux utilisateurs ou groupes qui nécessitent ces autorisations.

Rubriques

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

AWSCodeBuildAdminAccess

La `AWSCodeBuildAdminAccess` politique fournit un accès complet aux projets de CodeBuild construction CodeBuild, y compris les autorisations nécessaires à leur administration. Appliquez cette politique uniquement aux utilisateurs de niveau administratif afin de leur accorder un contrôle total sur les CodeBuild projets, les groupes de rapports et les ressources associées de votre AWS compte, y compris la possibilité de supprimer des projets et des groupes de rapports.

La stratégie `AWSCodeBuildAdminAccess` contient la déclaration suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "codebuild:*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "codecommit:ListRepositories",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeVpcs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "elasticfilesystem:DescribeFileSystems",
        "events>DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```

    },
    {
      "Action": [
        "logs:DeleteLogGroup"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:log-group:/aws/codebuild/*:log-stream:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": "arn:aws:ecs:*:*:task/*/*"
    },
    {
      "Sid": "CodeStarConnectionsReadWriteAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:UpdateConnectionInstallation",
        "codestar-connections:TagResource",
        "codestar-connections:UntagResource",
        "codestar-connections:ListConnections",
        "codestar-connections:ListInstallationTargets",
        "codestar-connections:ListTagsForResource",
        "codestar-connections:GetConnection",
        "codestar-connections:GetIndividualAccessToken",
        "codestar-connections:GetInstallationUrl",
        "codestar-connections:PassConnection",
        "codestar-connections:StartOAuthHandshake",
        "codestar-connections:UseConnection"
      ],
      "Resource": "arn:aws:codestar-connections:*:*:connection/*"
    },
    {

```

```
"Sid": "CodeStarNotificationsReadWriteAccess",
"Effect": "Allow",
"Action": [
  "codestar-notifications:CreateNotificationRule",
  "codestar-notifications:DescribeNotificationRule",
  "codestar-notifications:UpdateNotificationRule",
  "codestar-notifications>DeleteNotificationRule",
  "codestar-notifications:Subscribe",
  "codestar-notifications:Unsubscribe"
],
"Resource": "*",
"Condition": {
  "StringLike": {
    "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
  }
}
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
  "Effect": "Allow",
  "Action": [
    "sns:CreateTopic",
    "sns:SetTopicAttributes"
  ],
  "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics",
    "sns:GetTopicAttributes"
  ],
}
```



```
    "Resource": "*"
  },
  {
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }
]
```

AWSCodeBuildDeveloperAccess

La `AWSCodeBuildDeveloperAccess` politique permet d'accéder à toutes les fonctionnalités et aux ressources liées aux projets CodeBuild et aux groupes de rapports. Cette politique n'autorise pas les utilisateurs à supprimer CodeBuild des projets, des groupes de rapports ou des ressources connexes dans d'autres AWS services, tels que CloudWatch les événements. Nous vous recommandons d'appliquer cette stratégie à la plupart des utilisateurs.

La stratégie `AWSCodeBuildDeveloperAccess` contient la déclaration suivante :

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:StartBuildBatch",
        "codebuild:StopBuildBatch",
        "codebuild:RetryBuild",
        "codebuild:RetryBuildBatch",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
```

```

    "cloudwatch:GetMetricStatistics",
    "events:DescribeRule",
    "events:ListTargetsByRule",
    "events:ListRuleNamesByTarget",
    "logs:GetLogEvents",
    "s3:GetBucketLocation",
    "s3:ListAllMyBuckets"
  ],
  "Effect": "Allow",
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:PutParameter"
  ],
  "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
  "Sid": "CodeStarConnectionsUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-connections:ListConnections",
    "codestar-connections:GetConnection"
  ],
  "Resource": "arn:aws:codestar-connections:*:*:connection/*"
},
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
}

```

```
"Resource": "*",
"Condition": {
  "StringLike": {
    "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics",
    "sns:GetTopicAttributes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
  "Effect": "Allow",
  "Action": [
    "chatbot:DescribeSlackChannelConfigurations",
    "chatbot:ListMicrosoftTeamsChannelConfigurations"
  ],
  "Resource": "*"
}
],
"Version": "2012-10-17"
}
```

AWSCodeBuildReadOnlyAccess

La `AWSCodeBuildReadOnlyAccess` politique accorde un accès en lecture seule aux CodeBuild ressources associées dans d'autres AWS services. Appliquez cette stratégie aux utilisateurs qui peuvent afficher et exécuter des builds, afficher des projets et afficher des groupes de rapports, mais qui ne peuvent pas y apporter de modifications.

La stratégie `AWSCodeBuildReadOnlyAccess` contient la déclaration suivante :

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:List*",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "CodeStarConnectionsUserAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
      ],
      "Resource": "arn:aws:codestar-connections:*:*:connection/*"
    },
    {
      "Sid": "CodeStarNotificationsPowerUserAccess",
      "Effect": "Allow",
      "Action": [
```

```
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
    }
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}
],
"Version": "2012-10-17"
}
```

CodeBuild politiques et notifications gérées

CodeBuild prend en charge les notifications, qui peuvent informer les utilisateurs des modifications importantes apportées aux projets de construction. Les politiques gérées CodeBuild incluent des déclarations de politique relatives à la fonctionnalité de notification. Pour plus d'informations, consultez [En quoi consistent les notifications ?](#)

Autorisations liées aux notifications dans les stratégies gérées d'accès complet

La stratégie gérée `AWSCodeBuildFullAccess` inclut les déclarations suivantes pour permettre un accès complet aux notifications. Les utilisateurs auxquels cette politique gérée est appliquée peuvent également créer et gérer des sujets Amazon SNS pour les notifications, abonner et désinscrire des utilisateurs à des sujets, répertorier les sujets à choisir comme cibles pour les règles de notification et répertorier les AWS Chatbot clients configurés pour Slack.

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
```

```

        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
    "Effect": "Allow",
    "Action": [
        "sns:CreateTopic",
        "sns:SetTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",

```

```

    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }

```

Autorisations liées aux notifications dans les stratégies gérées en lecture seule

La stratégie gérée `AWSCodeBuildReadOnlyAccess` inclut les déclarations suivantes pour autoriser l'accès en lecture seule aux notifications. Les utilisateurs auxquels s'applique cette stratégie gérée peuvent voir des notifications pour les ressources, mais ne peuvent ni les créer, ni les gérer ni s'y abonner.

```

{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}

```

Autorisations liées aux notifications dans d'autres stratégies gérées

La stratégie gérée `AWSCodeBuildDeveloperAccess` inclut les déclarations suivantes pour autoriser les utilisateurs à créer et modifier des notifications, et s'y abonner. Les utilisateurs ne peuvent pas supprimer les règles de notification ni gérer les balises pour les ressources.

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
  "Effect": "Allow",
  "Action": [
    "chatbot:DescribeSlackChannelConfigurations",
    "chatbot:ListMicrosoftTeamsChannelConfigurations"
  ],
  "Resource": "*"
}
```


}

Pour plus d'informations sur l'IAM et les notifications, voir [Identity and Access ManagementAWS CodeStarNotifications](#).

CodeBuild mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CodeBuild depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au fil RSS sur [AWS CodeBuild Historique du document du guide de l'utilisateur](#).

| Modification | Description | Date |
|--|---|-------------|
| <code>AWSCodeBuildAdminAccess</code> et <code>AWSCodeBuildDeveloperAccess</code> — Mise à jour des politiques existantes | CodeBuild a ajouté une autorisation à ces politiques pour prendre en charge un type de notification supplémentaire en utilisant AWS Chatbot. Les <code>AWSCodeBuildDeveloperAccess</code> politiques <code>AWSCodeBuildAdminAccess</code> et ont été modifiées pour ajouter une autorisation, <code>chatbot:ListMicrosoftTeamsChannelConfigurations</code> . | 16 mai 2023 |
| CodeBuild a commencé à suivre les modifications | CodeBuild a commencé à suivre les modifications apportées AWS à ses politiques gérées. | 16 mai 2021 |

Exemples de politiques gérées par le client

Dans cette section, vous trouverez des exemples de stratégies utilisateur qui accordent des autorisations pour des actions AWS CodeBuild . Ces politiques fonctionnent lorsque vous utilisez l' CodeBuild API, AWS les SDK ou AWS CLI. Lorsque vous utilisez la console, vous devez accorder des autorisations supplémentaires spécifiques à la console. Pour plus d'informations, consultez [Autorisations requises pour utiliser la console AWS CodeBuild](#).

Vous pouvez utiliser les exemples de politiques IAM suivants pour limiter l' CodeBuild accès de vos utilisateurs et de vos rôles.

Rubriques

- [Autorisation d'un utilisateur à obtenir des informations sur les projets de génération](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les rapports](#)
- [Autorisation d'un utilisateur à créer des projets de génération](#)
- [Autorisation d'un utilisateur à créer un groupe de rapports](#)
- [Autorisation d'un utilisateur à supprimer un groupe de rapports](#)
- [Autorisation d'un utilisateur à supprimer un rapport](#)
- [Autorisation d'un utilisateur à supprimer des projets de génération](#)
- [Autorisation d'un utilisateur à obtenir une liste de noms de projet de génération](#)
- [Autorisation d'un utilisateur à modifier des informations sur les projets de génération](#)
- [Autorisation d'un utilisateur à modifier un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir des informations sur les générations](#)
- [Autorisation d'un utilisateur à obtenir une liste d'ID de génération pour un projet de génération](#)
- [Autorisation d'un utilisateur à obtenir une liste d'ID de génération](#)
- [Autorisation d'un utilisateur à obtenir une liste de groupes de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports](#)
- [Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport](#)
- [Autorisation d'un utilisateur à démarrer l'exécution de générations](#)
- [Autorisation d'un utilisateur à tenter d'arrêter des générations](#)
- [Autorisation d'un utilisateur à tenter de supprimer des générations](#)

- [Autorisation d'un utilisateur à obtenir des informations sur les images Docker gérées par CodeBuild](#)
- [Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC](#)
- [Utilisez une déclaration de refus pour éviter AWS CodeBuild de vous déconnecter des fournisseurs de sources](#)

Autorisation d'un utilisateur à obtenir des informations sur les projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur des projets de génération dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir des informations sur les groupes de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir des informations sur les groupes de rapports dans la région us-east-2 du compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir des informations sur les rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir des informations sur les rapports dans la région us-east-2 pour le compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à créer des projets de génération

L'exemple de déclaration de politique suivant permet à un utilisateur de créer des projets de construction avec n'importe quel nom, mais uniquement dans la us-east-2 région pour le compte 123456789012 et en utilisant uniquement le rôle de CodeBuild service spécifié :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
    }
  ]
}
```

L'exemple de déclaration de politique suivant permet à un utilisateur de créer des projets de construction avec n'importe quel nom, mais uniquement dans la us-east-2 région pour le compte 123456789012 et uniquement en utilisant le rôle de CodeBuild service spécifié. Cela garantit

également que l'utilisateur ne peut utiliser le rôle de service spécifié qu'avec AWS CodeBuild aucun autre AWS service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole",
      "Condition": {
        "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
      }
    }
  ]
}
```

Autorisation d'un utilisateur à créer un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de créer un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à supprimer un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de supprimer un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à supprimer un rapport

L'exemple suivant d'instruction de stratégie permet à un utilisateur de supprimer un rapport dans la région us-east-2 pour le compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReport",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à supprimer des projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à supprimer des projets de génération dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

```
]
}
```

Autorisation d'un utilisateur à obtenir une liste de noms de projet de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir une liste de noms de projet de génération pour le même compte :

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

Autorisation d'un utilisateur à modifier des informations sur les projets de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à modifier les informations sur des projets de génération avec n'importe quel nom, mais uniquement dans la région us-east-2 pour le compte 123456789012, et à utiliser uniquement le rôle de service AWS CodeBuild spécifié :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
    }
  ]
}
```

Autorisation d'un utilisateur à modifier un groupe de rapports

L'exemple suivant d'instruction de stratégie permet à un utilisateur de modifier un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir des informations sur les générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur les générations dans la région us-east-2 pour le compte 123456789012 pour les projets de génération nommés my-build-project et my-other-build-project :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetBuilds",
      "Resource": [
        "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
        "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
      ]
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir une liste d'ID de génération pour un projet de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir une liste d'ID de génération uniquement dans la région us-east-2 pour le compte 123456789012 pour les projets de génération nommés my-build-project et my-other-build-project :


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
      "Resource": [
        "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
        "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
      ]
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir une liste d'ID de génération

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir une liste de tous les ID de génération pour le même compte :

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

Autorisation d'un utilisateur à obtenir une liste de groupes de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de groupes de rapports dans la région us-east-2 pour le compte 123456789012 :

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

```
    }  
  ]  
}
```

Autorisation d'un utilisateur à obtenir une liste de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de rapports dans la région us-east-2 pour le compte 123456789012 :

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

Autorisation d'un utilisateur à obtenir une liste de rapports pour un groupe de rapports

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de rapports pour un groupe de rapports dans la région us-east-2 pour le compte 123456789012 :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "codebuild:ListReportsForReportGroup",  
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"  
    }  
  ]  
}
```

Autorisation d'un utilisateur à obtenir une liste de cas de test pour un rapport

L'exemple de déclaration de stratégie suivant permet à un utilisateur d'obtenir une liste de cas de test pour un rapport dans la région us-east-2 pour le compte 123456789012 :

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "codebuild:DescribeTestCases",
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
  }
]
```

Autorisation d'un utilisateur à démarrer l'exécution de générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à exécuter des générations dans la région `us-east-2` pour le compte `123456789012` pour les projets de génération dont le nom commence par `my` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Autorisation d'un utilisateur à tenter d'arrêter des générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à tenter d'arrêter des générations uniquement dans la région `us-east-2` pour le compte `123456789012` pour tout projet de génération dont le nom commence par `my` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

```
}
```

Autorisation d'un utilisateur à tenter de supprimer des générations

L'exemple de déclaration de stratégie suivant autorise un utilisateur à tenter de supprimer des générations uniquement dans la région us-east-2 pour le compte 123456789012 pour tout projet de génération dont le nom commence par my :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchDeleteBuilds",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Autorisation d'un utilisateur à obtenir des informations sur les images Docker gérées par CodeBuild

L'exemple de déclaration de stratégie suivant autorise un utilisateur à obtenir des informations sur toutes les images Docker gérées par CodeBuild :

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

Autoriser CodeBuild l'accès aux AWS services requis pour créer une interface réseau VPC

L'exemple de déclaration de politique suivant accorde AWS CodeBuild l'autorisation de créer une interface réseau dans un VPC avec deux sous-réseaux :

```
{
```

```

"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:region:account-id:network-interface/*",
    "Condition": {
      "StringEquals": {
        "ec2:AuthorizedService": "codebuild.amazonaws.com"
      },
      "ArnEquals": {
        "ec2:Subnet": [
          "arn:aws:ec2:region:account-id:subnet/subnet-id-1",
          "arn:aws:ec2:region:account-id:subnet/subnet-id-2"
        ]
      }
    }
  }
]
}

```

Utilisez une déclaration de refus pour éviter AWS CodeBuild de vous déconnecter des fournisseurs de sources

L'exemple de déclaration de stratégie suivant utilise une déclaration de refus pour empêcher AWS CodeBuild de se déconnecter des fournisseurs de source. Il utilise `codebuild:DeleteAuthToken`, qui est l'inverse de `codebuild:PersistAuthToken` et de `codebuild:ImportSourceCredentials`, pour se connecter aux fournisseurs de source. Pour

de plus amples informations, veuillez consulter [Autorisations requises pour que la AWS CodeBuild console puisse se connecter aux fournisseurs de sources](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "codebuild:DeleteOAuthToken",
      "Resource": "*"
    }
  ]
}
```

AWS CodeBuild référence aux autorisations

Vous pouvez utiliser des AWS clés de condition larges dans vos AWS CodeBuild polices pour exprimer des conditions. Pour obtenir une liste, consultez la section [Clés disponibles](#) dans le guide de l'utilisateur IAM.

Vous spécifiez les actions dans le champ `Action` de la politique. Pour spécifier une action, utilisez le préfixe `codebuild:` suivi du nom de l'opération d'API (par exemple, `codebuild:CreateProject` ou `codebuild:StartBuild`). Pour spécifier plusieurs actions dans une même instruction, séparez-les par une virgule (par exemple, `"Action": ["codebuild:CreateProject", "codebuild:StartBuild"]`).

Utilisation de caractères génériques

Vous spécifiez un ARN, avec ou sans caractère générique (*) comme valeur de ressource dans le champ `Resource` de la stratégie. Vous pouvez utiliser un caractère générique pour spécifier plusieurs actions ou ressources. Par exemple, `codebuild:*` spécifie toutes les CodeBuild actions et `codebuild:Batch*` indique toutes les CodeBuild actions qui commencent par le mot `Batch`. L'exemple suivant accorde l'accès à tous les projets de génération dont le nom commence par `my` :

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

CodeBuild Opérations d'API et autorisations requises pour les actions

BatchDeleteBuilds

Action : `codebuild:BatchDeleteBuilds`

Requête pour supprimer des générations

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetBuilds

Action : `codebuild:BatchGetBuilds`

Requête pour obtenir des informations sur des générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetProjects

Action : `codebuild:BatchGetProjects`

Requête pour obtenir des informations sur des projets de génération.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetReportGroups

Action : `codebuild:BatchGetReportGroups`

Obligatoire pour obtenir des informations sur les groupes de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchGetReports

Action : `codebuild:BatchGetReports`

Obligatoire pour obtenir des informations sur des générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchPutTestCases¹

Action : `codebuild:BatchPutTestCases`

Obligatoire pour créer ou mettre à jour un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateProject

Actions : `codebuild:CreateProject`, `iam:PassRole`

Requise pour créer des projets de génération.

Ressources :

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

CreateReport¹

Action : `codebuild:CreateReport`

Obligatoire pour créer un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateReportGroup

Action : `codebuild:CreateReportGroup`

Obligatoire pour créer un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateWebhook

Action : `codebuild:CreateWebhook`

Requise pour créer un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteProject

Action : `codebuild>DeleteProject`

Nécessaire pour supprimer un CodeBuild projet.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteReport

Action : `codebuild>DeleteReport`

Obligatoire pour supprimer une règle.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteReportGroup

Action : `codebuild>DeleteReportGroup`

Obligatoire pour supprimer un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteSourceCredentials

Action : `codebuild>DeleteSourceCredentials`

Nécessaire pour supprimer un ensemble d'`SourceCredentialsInfo`objets contenant des informations sur les informations d'identification d'un GitHub référentiel GitHub Enterprise Server ou Bitbucket.

Ressource : *

DeleteWebhook

Action : `codebuild>DeleteWebhook`

Requise pour créer un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DescribeTestCases

Action : `codebuild>DescribeTestCases`

Obligatoire pour renvoyer une liste paginée de cas de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

ImportSourceCredentials

Action : `codebuild:ImportSourceCredentials`

Nécessaire pour importer un ensemble d'`SourceCredentialsInfo`objets contenant des informations sur les informations d'identification d'un GitHub référentiel GitHub Enterprise Server ou Bitbucket.

Ressource : *

InvalidateProjectCache

Action : `codebuild:InvalidateProjectCache`

Requise pour réinitialiser le cache pour un projet.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuildBatches

Action : `codebuild:ListBuildBatches`

Nécessaire pour obtenir une liste des identifiants de lots de build.

Ressource : *

ListBuildBatchesForProject

Action : `codebuild:ListBuildBatchesForProject`

Nécessaire pour obtenir une liste des identifiants de lots de construction pour un projet spécifique.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuilds

Action : `codebuild:ListBuilds`

Requise pour obtenir une liste d'ID de génération.

Ressource : *

ListBuildsForProject

Action : `codebuild:ListBuildsForProject`

Requise pour obtenir une liste d'ID pour un projet de génération.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListCuratedEnvironmentImages

Action : `codebuild:ListCuratedEnvironmentImages`

Requise pour obtenir des informations sur toutes les images Docker gérées par AWS CodeBuild.

Ressource : * (requise, mais ne fait pas référence à une ressource AWS adressable)

ListProjects

Action : `codebuild:ListProjects`

Requise pour obtenir une liste de noms de projet de génération.

Ressource : *

ListReportGroups

Action : `codebuild:ListReportGroups`

Obligatoire pour obtenir une liste des groupes de rapports.

Ressource : *

ListReports

Action : `codebuild:ListReports`

Obligatoire pour obtenir une liste de rapports.

Ressource : *

ListReportsForReportGroup

Action : `codebuild:ListReportsForReportGroup`

Obligatoire pour obtenir une liste de rapports pour un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

RetryBuild

Action : `codebuild:RetryBuild`

Nécessaire pour réessayer les builds.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StartBuild

Action : `codebuild:StartBuild`

Requise pour démarrer l'exécution de générations.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StopBuild

Action : `codebuild:StopBuild`

Requise pour tenter d'arrêter des générations en cours d'exécution.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

UpdateProject

Actions : `codebuild:UpdateProject`, `iam:PassRole`

Requise pour modifier des informations sur les générations.

Ressources :

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateProjectVisibility

Actions : `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Nécessaire pour modifier la visibilité publique des versions d'un projet.

Ressources :

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`

- `arn:aws:iam::account-ID:role/role-name`

UpdateReport¹

Action : `codebuild:UpdateReport`

Obligatoire pour créer ou mettre à jour un rapport de test.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateReportGroup

Action : `codebuild:UpdateReportGroup`

Obligatoire pour mettre à jour un groupe de rapports.

Ressource : `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateWebhook

Action : `codebuild:UpdateWebhook`

Requise pour mettre à jour un webhook.

Ressource : `arn:aws:codebuild:region-ID:account-ID:project/project-name`

¹ Utilisé uniquement à des fins d'autorisation. Il n'existe aucune API pour cette action.

Utilisation de balises pour contrôler l'accès aux ressources AWS CodeBuild

Les conditions figurant dans les déclarations de politique IAM font partie de la syntaxe que vous pouvez utiliser pour spécifier les autorisations relatives aux actions basées sur des CodeBuild projets. Vous pouvez créer une politique qui autorise ou refuse des actions sur des projets en fonction des balises associées à ces projets, puis appliquer ces politiques aux groupes IAM que vous configurez pour gérer les utilisateurs. Pour plus d'informations sur l'application de balises à un projet à l'aide de la console ou AWS CLI consultez [Création d'un projet de génération dans AWS CodeBuild](#). Pour plus d'informations sur l'application de balises à l'aide du CodeBuild SDK, consultez [CreateProject](#) la section « [Tags](#) » dans le manuel de référence des CodeBuild API. Pour plus d'informations sur l'utilisation de balises pour contrôler l'accès aux AWS ressources, consultez la section [Contrôle de l'accès aux AWS ressources à l'aide de balises de ressources](#) dans le guide de l'utilisateur IAM.

Exemple Exemple 1 : Limiter les actions CodeBuild du projet en fonction des balises de ressources

L'exemple suivant refuse toutes les actions `BatchGetProjects` sur les projets marqués avec la clé `Environment` utilisant la valeur `Production`. L'administrateur d'un utilisateur doit associer cette politique IAM en plus de la politique utilisateur gérée aux utilisateurs non autorisés. La clé de condition `aws:ResourceTag` est utilisée pour contrôler l'accès aux ressources en fonction de leurs balises.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetProjects"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:ResourceTag/Environment": "Production"
        }
      }
    }
  ]
}
```

Exemple Exemple 2 : Limiter les actions CodeBuild du projet en fonction des balises de demande

La stratégie suivante refuse aux utilisateurs l'autorisation pour l'action `CreateProject` si la demande contient une balise avec la clé `Environment` et la valeur de clé `Production`. En outre, la stratégie empêche ces utilisateurs non autorisés de modifier des projets en utilisant la clé de condition `aws:TagKeys` pour ne pas autoriser `UpdateProject` si la requête contient une balise avec la clé `Environment`. Un administrateur doit associer cette politique IAM en plus de la politique utilisateur gérée aux utilisateurs qui ne sont pas autorisés à effectuer ces actions. La clé de `aws:RequestTag` condition est utilisée pour contrôler les balises qui peuvent être transmises dans une demande IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Effect": "Deny",
    "Action": [
      "codebuild:CreateProject"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:RequestTag/Environment": "Production"
      }
    }
  },
  {
    "Effect": "Deny",
    "Action": [
      "codebuild:UpdateProject"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": ["Environment"]
      }
    }
  }
]
```

Exemple Exemple 3 : Refuser ou autoriser des actions sur des groupes de rapports en fonction des balises de ressources

Vous pouvez créer une politique qui autorise ou refuse les actions sur les CodeBuild ressources (projets et groupes de rapports) en fonction des AWS balises associées à ces ressources, puis appliquer ces politiques aux groupes IAM que vous configurez pour gérer les utilisateurs. Par exemple, vous pouvez créer une politique qui refuse toutes les CodeBuild actions sur tout groupe de rapports dont la clé de AWS balise Status et la valeur clé sont Secret, puis appliquer cette politique au groupe IAM que vous avez créé pour les développeurs généraux (*développeurs*). Vous devez ensuite vous assurer que les développeurs travaillant sur ces groupes de rapports balisés ne sont pas membres de ce groupe général de *développeurs*, mais appartiennent plutôt à un autre groupe IAM auquel la politique restrictive n'est pas appliquée (SecretDevelopers).

L'exemple suivant refuse toutes les CodeBuild actions sur les groupes de rapports marqués par la clé Status et la valeur clé de Secret :

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild>DeleteReportGroup",
        "codebuild:ListReportGroups",
        "codebuild:ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
      ]
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : "aws:ResourceTag/Status": "Secret"
      }
    }
  ]
}
```

Exemple Exemple 4 : Limiter CodeBuild les actions en `AWSCodeBuildDeveloperAccess` fonction des balises de ressources

Vous pouvez créer des politiques qui autorisent CodeBuild des actions sur tous les groupes de rapports et projets qui ne sont pas marqués par des balises spécifiques. Par exemple, la stratégie suivante accorde l'équivalent des autorisations [AWSCodeBuildDeveloperAccess](#) pour tous les groupes de rapports, sauf ceux balisés avec les balises spécifiées :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:List*",
        "codecommit:GetBranch",

```



```
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*",
    "Condition": {
        "StringNotEquals": {
            "aws:ResourceTag/Status": "Secret",
            "aws:ResourceTag/Team": "Saanvi"
        }
    }
}
]
```

Affichage des ressources dans la console

La AWS CodeBuild console nécessite l'`ListRepositories` autorisation d'afficher la liste des référentiels pour votre AWS compte dans la AWS région où vous êtes connecté. La console comprend également une fonction Go to resource (Accéder aux ressources) qui permet d'effectuer rapidement une recherche de ressources sensible à la casse. Cette recherche est effectuée dans votre AWS compte dans la AWS région où vous êtes connecté. Les ressources suivantes sont affichées pour les services suivants :

- AWS CodeBuild : Projets de génération
- AWS CodeCommit : Référentiels
- AWS CodeDeploy : Applications
- AWS CodePipeline : Pipelines

Pour effectuer cette recherche pour les ressources dans tous les services, vous devez disposer des autorisations suivantes :

- CodeBuild: `ListProjects`

- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Les résultats ne sont pas renvoyés pour les ressources d'un service si vous ne disposez pas d'autorisations pour ce service. Même si vous êtes autorisé à afficher des ressources, certaines ressources ne sont pas renvoyées si une valeur Deny explicite est définie pour l'affichage de ces ressources.

Validation de conformité pour AWS CodeBuild

Des auditeurs tiers évaluent la sécurité et AWS CodeBuild la conformité de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformité](#). Pour obtenir des informations générales, veuillez consulter [Programmes de conformité d'AWS](#).

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, consultez la section [Téléchargement de rapports dans AWS Artifact](#).

Votre responsabilité en matière de conformité lors de l'utilisation CodeBuild est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Si votre utilisation de CodeBuild est soumise à la conformité à des normes telles que HIPAA, PCI ou FedRAMP, fournir des ressources pour vous aider à : AWS

- [Guides de démarrage rapide sur la sécurité et la conformité](#) : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS
- Livre blanc [sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc](#) décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- [AWS ressources de conformité](#) — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [AWS Config](#)— Ce AWS service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)— Surveillez votre utilisation AWS CodeBuild en ce qui concerne les meilleures pratiques de sécurité en utilisant [AWS Security Hub](#). Security Hub utilise des contrôles de sécurité

pour évaluer les configurations des ressources et les normes de sécurité afin de vous aider à respecter divers cadres de conformité. Pour plus d'informations sur l'utilisation de Security Hub pour évaluer les CodeBuild ressources, consultez la section [AWS CodeBuild Contrôles](#) du Guide de AWS Security Hub l'utilisateur.

Résilience dans AWS CodeBuild

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

Sécurité de l'infrastructure dans AWS CodeBuild

En tant que service géré, AWS CodeBuild il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder CodeBuild via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#)

(AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Accédez à votre fournisseur source dans CodeBuild

Pour GitHub ou GitHub Enterprise Server, vous utilisez un jeton d'accès personnel pour accéder au fournisseur source. Pour Bitbucket, vous utilisez un mot de passe d'application pour accéder au fournisseur source.

Rubriques

- [GitHub et jeton d'accès au serveur GitHub Enterprise](#)
- [Mot de passe de l'application Bitbucket](#)

GitHub et jeton d'accès au serveur GitHub Enterprise

Prérequis de jeton d'accès

Avant de commencer, vous devez ajouter les étendues d'autorisation appropriées à votre jeton GitHub d'accès.

En effet GitHub, votre jeton d'accès personnel doit avoir les champs d'application suivants.

- `repo` : accorde un contrôle total des référentiels privés.
- `repo:status` : accorde un accès en lecture/écriture aux statuts de validation des référentiels publics et privés.
- `admin:repo_hook` : accorde le contrôle total des hooks de référentiel. Cette règle n'est pas obligatoire si votre jeton dispose de la règle `repo`.

Pour plus d'informations, consultez la section [Comprendre les étendues des applications OAuth](#) sur le site Web. GitHub

Connect GitHub avec un jeton d'accès (console)

Pour utiliser la console afin de connecter votre projet à GitHub un jeton d'accès, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Dans le champ Source provider, sélectionnez GitHub.
2. Pour Repository, choisissez Connect with a GitHub personal access token.
3. Dans jeton d'accès GitHub personnel, entrez votre jeton d'accès GitHub personnel.
4. Choisissez Save token (Enregistrer le jeton).

Connectez-vous à GitHub l'aide d'un jeton d'accès (CLI)

Suivez ces étapes pour connecter votre projet AWS CLI à l' GitHub aide d'un jeton d'accès. Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

1. Exécutez la commande import-source-credentials :

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, *import-source-credentials.json*) à un emplacement de l'ordinateur local ou de l'instance sur laquelle le AWS CLI est installé. Modifiez les données copiées comme suit, puis enregistrez vos résultats.

```
{
  "serverType": "server-type",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Remplacez les éléments suivants :

- *server-type* : valeur obligatoire. Le fournisseur de source utilisé pour cette information d'identification. Les valeurs valides sont GITHUB ou GITHUB_ENTERPRISE.
- *auth-type* : valeur obligatoire. Type d'authentification utilisé pour se connecter à un référentiel GitHub ou à un référentiel GitHub Enterprise Server. Les valeurs valides comprennent PERSONAL_ACCESS_TOKEN et BASIC_AUTH. Vous ne pouvez pas utiliser l' CodeBuild API pour créer une connexion OAUTH. Vous devez plutôt utiliser la CodeBuild console.

- *should-override* : valeur facultative. Définissez cette valeur sur `false` pour empêcher le remplacement des informations d'identification source du référentiel. Définissez cette valeur sur `true` pour remplacer les informations d'identification source du référentiel. La valeur par défaut est `true`.
 - *token* : valeur obligatoire. Pour GitHub notre serveur d' GitHub entreprise, il s'agit du jeton d'accès personnel.
 - *username* : valeur facultative. Ce paramètre est ignoré pour GitHub les fournisseurs de sources GitHub Enterprise Server.
2. Pour vous connecter à votre compte avec un jeton d'accès, passez au répertoire contenant le fichier `import-source-credentials.json` que vous avez enregistré à l'étape 1 et exécutez à nouveau la commande `import-source-credentials`.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Les données au format JSON apparaissent dans la sortie avec un Amazon Resource Name (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Si vous exécutez la commande `import-source-credentials` avec le même type de serveur et de auth une seconde fois, le jeton d'accès stocké est mis à jour.

Une fois que votre compte est connecté par un jeton d'accès, vous pouvez l'utiliser `create-project` pour créer votre CodeBuild projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

3. Pour afficher les jetons d'accès connectés, exécutez la commande `list-source-credentials`.

```
aws codebuild list-source-credentials
```

Un objet `sourceCredentialsInfos` au format JSON apparaît dans la sortie :

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "server-type",
      "arn": "arn"
    }
  ]
}
```

Le `sourceCredentialsObject` contient la liste des informations d'identification de la source connectée :

- L'élément `authType` est le type d'authentification utilisé par les informations d'identification. Cette valeur peut être `OAUTH`, `BASIC_AUTH` ou `PERSONAL_ACCESS_TOKEN`.
 - Le `serverType` est le type de fournisseur de source. Cette valeur peut être `GITHUB`, `GITHUB_ENTERPRISE` ou `BITBUCKET`.
 - Le `arn` est l'ARN du jeton.
4. Pour vous déconnecter d'un fournisseur de source et supprimer ses jetons d'accès, exécutez la commande `delete-source-credentials` avec son ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Des données au format JSON sont renvoyées avec un ARN des informations d'identification supprimées.

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Mot de passe de l'application Bitbucket

Exigences relatives au mot de passe de l'application

Avant de commencer, vous devez ajouter les étendues d'autorisation appropriées au mot de passe de votre application Bitbucket.

Pour Bitbucket, votre mot de passe d'application doit disposer des règles suivantes.

- `repository:read` : accorde l'accès en lecture à tous les référentiels auxquels l'utilisateur d'autorisation a accès.
- `pullrequest:read` : accorde l'accès en lecture aux demandes d'extraction. Si votre projet Bitbucket possède un webhook, le mot de passe doit disposer de cette règle.
- `webhook` : accorde l'accès aux webhooks. Si votre projet possède une opération webhook, alors le mot de passe doit disposer de cette règle.

Pour plus d'informations, consultez [Règles pour l'API REST Bitbucket Cloud](#) et [OAuth sur Bitbucket Cloud](#).

Connect Bitbucket à l'aide d'un mot de passe d'application (console)

Pour utiliser la console afin de connecter votre projet à Bitbucket à l'aide d'un mot de passe d'application, procédez comme suit lorsque vous créez un projet. Pour plus d'informations, consultez [Création d'un projet de génération \(console\)](#).

1. Pour Source provider (Fournisseur de source), choisissez Bitbucket.

Note

CodeBuild ne prend pas en charge Bitbucket Server.

2. Pour Repository (Référentiel), choisissez Connect with a Bitbucket app password (Se connecter avec un mot de passe d'application Bitbucket).
3. Dans Bitbucket username (Nom d'utilisateur Bitbucket), saisissez votre nom d'utilisateur Bitbucket.
4. Dans Bitbucket app password (Mot de passe d'application Bitbucket), saisissez votre mot de passe d'application Bitbucket.
5. Choisissez Save Bitbucket credentials (Enregistrer les informations d'identification Bitbucket).

Connect Bitbucket avec un mot de passe d'application (CLI)

Suivez ces étapes pour connecter votre projet AWS CLI à Bitbucket à l'aide d'un mot de passe d'application. Pour plus d'informations sur l'utilisation du AWS CLI with AWS CodeBuild, consultez le [Référence des commandes en ligne](#).

1. Exécutez la commande `import-source-credentials` :

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Des données au format JSON apparaissent dans la sortie. Copiez les données dans un fichier (par exemple, `import-source-credentials.json`) à un emplacement de l'ordinateur local ou de l'instance sur laquelle le AWS CLI est installé. Modifiez les données copiées comme suit, puis enregistrez vos résultats.

```
{
  "serverType": "BITBUCKET",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Remplacez les éléments suivants :

- *auth-type* : valeur obligatoire. Type d'authentification utilisé pour se connecter à un dépôt Bitbucket. Les valeurs valides comprennent `PERSONAL_ACCESS_TOKEN` et `BASIC_AUTH`. Vous ne pouvez pas utiliser l' API CodeBuild pour créer une connexion OAUTH. Vous devez utiliser la console CodeBuild à la place.
 - *should-overwrite* : valeur facultative. Définissez cette valeur sur `false` pour empêcher le remplacement des informations d'identification source du référentiel. Définissez cette valeur sur `true` pour remplacer les informations d'identification source du référentiel. La valeur par défaut est `true`.
 - *token* : valeur obligatoire. Pour Bitbucket, il s'agit du mot de passe d'application.
 - *username* : valeur facultative. Le nom d'utilisateur Bitbucket `authType` est quand `BASIC_AUTH`. Ce paramètre est ignoré pour d'autres types de fournisseurs de source ou connexions.
2. Pour connecter votre compte à l'aide d'un mot de passe d'application, accédez au répertoire contenant le `import-source-credentials.json` fichier que vous avez enregistré à l'étape 1 et réexécutez la `import-source-credentials` commande.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Les données au format JSON apparaissent dans la sortie avec un Amazon Resource Name (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Si vous exécutez la commande `import-source-credentials` avec le même type de serveur et de auth une seconde fois, le jeton d'accès stocké est mis à jour.

Une fois que votre compte est connecté à l'aide d'un mot de passe d'application, vous pouvez `create-project` l'utiliser pour créer votre CodeBuild projet. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération \(AWS CLI\)](#).

3. Pour afficher les mots de passe des applications connectées, exécutez la `list-source-credentials` commande.

```
aws codebuild list-source-credentials
```

Un objet `sourceCredentialsInfos` au format JSON apparaît dans la sortie :

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "BITBUCKET",
      "arn": "arn"
    }
  ]
}
```

Le `sourceCredentialsObject` contient la liste des informations d'identification de la source connectée :

- L'élément `authType` est le type d'authentification utilisé par les informations d'identification. Cette valeur peut être `OAUTH`, `BASIC_AUTH` ou `PERSONAL_ACCESS_TOKEN`.

- Le arn est l'ARN du jeton.
4. Pour vous déconnecter d'un fournisseur source et supprimer le mot de passe de son application, exécutez la `delete-source-credentials` commande avec son ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Des données au format JSON sont renvoyées avec un ARN des informations d'identification supprimées.

```
{  
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"  
}
```

Prévention du cas de figure de l'adjoint désorienté entre services

Le problème de l'adjoint confus est un problème de sécurité dans lequel une entité qui n'a pas l'autorisation d'effectuer une action peut contraindre une entité plus privilégiée à effectuer cette action. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS CodeBuild accordent un autre service à la ressource. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

Le moyen le plus efficace de se protéger du problème de l'adjoint désorienté consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:codebuild:*:123456789012:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations.

La valeur de `aws:SourceArn` doit être l'ARN CodeBuild du projet.

L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale CodeBuild pour éviter le problème de confusion des adjoints.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
        }
      }
    }
  ]
}
```

Rubriques avancées

Cette section inclut plusieurs rubriques avancées qui s'avèrent utiles pour des utilisateurs AWS CodeBuild plus expérimentés.

Rubriques

- [Configuration avancée](#)
- [Référence des commandes en ligne pour AWS CodeBuild](#)
- [Référence des kits SDK et des outils AWS pour AWS CodeBuild](#)
- [Spécification du point de terminaison AWS CodeBuild](#)
- [Utilisation d'AWS CodePipeline avec AWS CodeBuild pour tester le code et exécuter des générations](#)
- [Utilisation d'AWS CodeBuild avec Jenkins](#)
- [Utilisation d'AWS CodeBuild avec Codecov](#)
- [Utiliser AWS CodeBuild avec applications sans serveur](#)

Configuration avancée

Si vous suivez les étapes décrites dans [Mise en route à l'aide de la console](#) pour accéder à AWS CodeBuild pour la première fois, vous n'avez probablement pas besoin des informations de cette rubrique. Toutefois, au fur et à mesure que vous continuez à utiliser CodeBuild, vous souhaiterez peut-être, par exemple, autoriser les groupes IAM et les utilisateurs de votre organisation à accéder à IAM CodeBuild, modifier les rôles de service existants dans IAM ou AWS KMS keys to access CodeBuild, ou les configurer AWS CLI sur l'ensemble des postes de travail de votre organisation pour y accéder CodeBuild. Cette rubrique explique comment effectuer les étapes de configuration associées.

Nous supposons que vous disposez déjà d'un compte AWS. Toutefois, si vous n'en possédez pas déjà un, rendez-vous [sur http://aws.amazon.com](http://aws.amazon.com), choisissez Se connecter à la console et suivez les instructions en ligne.

Rubriques

- [Ajouter des autorisations d' CodeBuild accès à un groupe ou à un utilisateur IAM](#)
- [Création d'un rôle CodeBuild de service](#)

- [Création et configuration d'une clé gérée par le client pour CodeBuild](#)
- [Installation et configuration de l'AWS CLI](#)

Ajouter des autorisations d' CodeBuild accès à un groupe ou à un utilisateur IAM

Pour accéder AWS CodeBuild à un groupe ou à un utilisateur IAM, vous devez ajouter des autorisations d'accès. Cette section explique comment procéder avec la console IAM ou le AWS CLI.

Si vous souhaitez accéder CodeBuild à votre compte AWS root (non recommandé) ou à l'aide d'un utilisateur administrateur de votre AWS compte, vous n'avez pas besoin de suivre ces instructions.

Pour plus d'informations sur les comptes AWS root et les utilisateurs administrateurs, voir [L'utilisateur Compte AWS root](#) et [Création de votre premier utilisateur Compte AWS root et de votre premier groupe](#) dans le guide de l'utilisateur.

Pour ajouter des autorisations d' CodeBuild accès à un groupe ou à un utilisateur IAM (console)

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Vous devez vous être déjà connecté à l'AWS Management Console à l'aide de l'une des actions suivantes :

- Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, voir [L'utilisateur Compte AWS root](#) dans le Guide de l'utilisateur.
- Un utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez [la section Création de votre premier utilisateur et groupe Compte AWS root](#) dans le Guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam>ListAttachedGroupPolicies
iam>ListAttachedUserPolicies
iam>ListGroups
iam>ListPolicies
iam>ListUsers
```

Pour plus d'informations, consultez la section [Présentation des politiques IAM](#) dans le guide de l'utilisateur.

2. Dans le panneau de navigation, choisissez Politiques (Politiques).
3. Pour ajouter un ensemble personnalisé d'autorisations d'AWS CodeBuild accès à un groupe IAM ou à un utilisateur IAM, passez de l'étape 4 de cette procédure.

Pour ajouter un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe IAM ou à un utilisateur IAM, choisissez Type de politique, AWSGéré, puis procédez comme suit :

- Pour ajouter des autorisations d'accès complètes à CodeBuild, cochez la case nommée `AWSCodeBuildAdminAccess`, choisissez Policy Actions, puis choisissez Joindre. Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette procédure pour les politiques nommées `AmazonS3ReadOnlyAccess` et `IAMFullAccess`.
- Pour ajouter des autorisations d'accès à CodeBuild tout, sauf à l'administration du projet de construction, cochez la case nommée `AWSCodeBuildDeveloperAccess`, choisissez Policy Actions, puis choisissez Attacher. Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette opération pour la politique nommée `AmazonS3ReadOnlyAccess`.
- Pour ajouter des autorisations d'accès en lecture seule à CodeBuild, sélectionnez les cases nommées `AWSCodeBuildReadOnlyAccess`. Cochez la case à côté du groupe ou de l'utilisateur IAM cible, puis choisissez Attach Policy. Répétez cette opération pour la politique nommée `AmazonS3ReadOnlyAccess`.

Vous avez maintenant ajouté un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe ou à un utilisateur IAM. Ignorez les autres étapes de cette procédure.

4. Choisissez Create Policy (Créer une politique).
5. Sur la page Créer une stratégie, choisissez Sélectionner en regard de Créer votre propre stratégie.
6. Sur la page Review Policy (Examiner la stratégie), pour Policy Name (Nom de la stratégie), saisissez un nouveau nom de stratégie (par exemple, **CodeBuildAccessPolicy**). Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.
7. Dans Policy Document (Document de stratégie), saisissez ce qui suit, puis sélectionnez Create Policy (Créer une stratégie).

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "CodeBuildAccessPolicy",
    "Effect": "Allow",
    "Action": [
      "codebuild:*"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeBuildRolePolicy",
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::account-ID:role/role-name"
  },
  {
    "Sid": "CloudWatchLogsAccessPolicy",
    "Effect": "Allow",
    "Action": [
      "logs:FilterLogEvents",
      "logs:GetLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3AccessPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:GetObject",
      "s3:List*",
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ]
  }
]
```



```
    ],  
    "Resource": "*"    
  }  
]  
}
```

Note

Cette politique permet d'accéder à toutes les CodeBuild actions et à un nombre potentiellement important de AWS ressources. Pour restreindre les autorisations à des CodeBuild actions spécifiques, modifiez la valeur de `codebuild:*` dans la déclaration de CodeBuild politique. Pour plus d'informations, veuillez consulter [Gestion des identités et des accès](#). Pour limiter l'accès à certaines ressources AWS, modifiez la valeur de l'objet Resource. Pour plus d'informations, veuillez consulter [Gestion des identités et des accès](#).

La `CodeBuildRolePolicy` déclaration est requise pour autoriser la création ou la modification d'un projet de construction.

8. Dans le volet de navigation, choisissez Groupes ou Utilisateurs.
9. Dans la liste des groupes ou des utilisateurs, choisissez le nom du groupe IAM ou de l'utilisateur IAM auquel vous souhaitez ajouter des autorisations CodeBuild d'accès.
10. Pour un groupe, sur la page des paramètres du groupe, dans l'onglet Permissions (Autorisations), développez Managed Policies (Stratégies gérées), puis choisissez Attach Policy (Attacher une stratégie).

Pour un utilisateur, sur la page des paramètres de l'utilisateur, dans l'onglet Autorisations, choisissez Ajouter des autorisations.

11. Pour un groupe, sur la page Joindre une politique, sélectionnez CodeBuildAccessPolicy, puis choisissez Joindre une politique.

Pour un utilisateur, sur la page Ajouter des autorisations, choisissez Joindre directement des politiques existantes. Sélectionnez CodeBuildAccessPolicy, choisissez Suivant : Révision, puis choisissez Ajouter des autorisations.

Pour ajouter des autorisations CodeBuild d'accès à un groupe ou à un utilisateur IAM (AWS CLI)

1. Assurez-vous d'avoir configuré le AWS CLI avec la clé AWS d'accès et la clé d'accès AWS secrète qui correspondent à l'une des entités IAM, comme décrit dans la procédure précédente. Pour plus d'informations, consultez [Préparation de l'installation de l'AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface.
2. Pour ajouter un ensemble personnalisé d'autorisations d'AWS CodeBuild accès à un groupe IAM ou à un utilisateur IAM, passez à l'étape 3 de cette procédure.

Pour ajouter un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe IAM ou à un utilisateur IAM, procédez comme suit :

Exécutez l'une des commandes suivantes, selon que vous souhaitez ajouter des autorisations à un groupe ou à un utilisateur IAM :

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn  
aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Vous devez exécuter la commande trois fois, en remplaçant le *nom du groupe* ou le *nom d'utilisateur par* le nom du groupe IAM ou le nom d'utilisateur, et en remplaçant *policy-arn* une fois pour chacune des politiques Amazon Resource Names (ARN) suivantes :

- Pour ajouter des autorisations d'accès complètes à CodeBuild, utilisez les ARN de politique suivants :
 - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
 - `arn:aws:iam::aws:policy/IAMFullAccess`
- Pour ajouter des autorisations d'accès à tous CodeBuild les domaines, à l'exception de l'administration du projet de construction, utilisez les règles ARN suivantes :
 - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Pour ajouter des autorisations d'accès en lecture seule à CodeBuild, utilisez les ARN de politique suivants :
 - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Vous avez maintenant ajouté un ensemble d'autorisations d' CodeBuild accès par défaut à un groupe ou à un utilisateur IAM. Ignorez les autres étapes de cette procédure.

3. Dans un répertoire vide sur le poste de travail ou l'instance local où l'AWS CLI est installée, créez un fichier nommé `put-group-policy.json` ou `put-user-policy.json`. Si vous choisissez un autre nom de fichier, veillez à l'utiliser tout au long de cette procédure.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "codebuild:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeBuildRolePolicy",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-ID:role/role-name"
    },
    {
      "Sid": "CloudWatchLogsAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3AccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:GetObject",
        "s3:List*"
      ],

```

```
    "s3:PutObject"
  ],
  "Resource": "*"
},
{
  "Sid": "S3BucketIdentity",
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketAcl",
    "s3:GetBucketLocation"
  ],
  "Resource": "*"
}
]
```

Note

Cette politique permet d'accéder à toutes les CodeBuild actions et à un nombre potentiellement important de AWS ressources. Pour restreindre les autorisations à des CodeBuild actions spécifiques, modifiez la valeur `codebuild:*` dans la déclaration de CodeBuild politique. Pour plus d'informations, veuillez consulter [Gestion des identités et des accès](#). Pour limiter l'accès à certaines ressources AWS, modifiez la valeur de l'objet `Resource` associé. Pour plus d'informations, consultez [Gestion des identités et des accès](#) ou la documentation de sécurité du service AWS spécifique.

La `CodeBuildRolePolicy` déclaration est requise pour autoriser la création ou la modification d'un projet de construction.

4. Placez-vous dans le répertoire dans lequel vous avez enregistré le fichier, puis exécutez l'une des commandes suivantes. Vous pouvez utiliser des valeurs différentes pour `CodeBuildGroupAccessPolicy` et `CodeBuildUserAccessPolicy`. Si vous utilisez des valeurs différentes, substituez-les ici.

Pour un groupe IAM :

```
aws iam put-group-policy --group-name group-name --policy-name
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Pour un utilisateur :

```
aws iam put-user-policy --user-name user-name --policy-name  
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

Dans les commandes précédentes, remplacez le *nom du groupe* ou le *nom d'utilisateur* *par* le nom du groupe ou de l'utilisateur IAM cible.

Création d'un rôle CodeBuild de service

Vous avez besoin d'un rôle de service AWS CodeBuild pour que CodeBuild puisse interagir avec les services AWS dépendants en votre nom. Vous pouvez créer un rôle CodeBuild de service à l'aide des consoles CodePipeline ou CodeBuild. Pour plus d'informations, consultez :

- [Création d'un projet de génération \(console\)](#)
- [Création d'un pipeline utilisant CodeBuild \(console CodePipeline\)](#)
- [Ajout d'une action de génération CodeBuild à un pipeline \(console CodePipeline\)](#)
- [Modification des paramètres d'un projet de génération \(console\)](#)

Si vous ne prévoyez pas d'utiliser ces consoles, cette section explique comment créer un rôle de service CodeBuild avec la console IAM ou le CLI AWS.

Important

CodeBuild utilise le rôle de service pour toutes les opérations effectuées en votre nom. Si le rôle comprend des autorisations que l'utilisateur ne devrait pas avoir, vous avez peut-être remonté accidentellement ses autorisations. Vérifiez que le rôle accorde le [privilège le plus faible](#).

Le rôle de service décrit sur cette page contient une stratégie qui accorde les autorisations minimales requises pour utiliser CodeBuild. Il se peut que vous ayez besoin d'ajouter des autorisations supplémentaires, selon votre cas d'utilisation.

Pour créer un rôle CodeBuild de service (console)

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Vous devez vous être déjà connecté à la console à l'aide de l'une des actions suivantes :

- Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, voir [L'utilisateurCompte AWS root](#) dans le Guide de l'utilisateur.
- Un utilisateur administrateur de votreAWS compte. Pour plus d'informations, consultez [la section Création de votre premier utilisateur et groupeCompte AWS root](#) dans le Guide de l'utilisateur.
- Un utilisateur de votreAWS compte autorisé à effectuer les actions minimales suivantes :

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam>ListAttachedRolePolicies
iam>ListPolicies
iam>ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Pour plus d'informations, consultez la section [Présentation des politiques IAM](#) dans le guide de l'utilisateur.

2. Dans le panneau de navigation, choisissez Politiques (Politiques).
3. Choisissez Create Policy (Créer une politique).
4. Sur la page Créer une stratégie, choisissez JSON.
5. Pour la stratégie JSON, saisissez ce qui suit, puis choisissez Review Policy (Examiner une stratégie) :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    }
  ]
}
```

```
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeCommitPolicy",
    "Effect": "Allow",
    "Action": [
      "codecommit:GitPull"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3GetObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3PutObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ECRPullPolicy",
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ECRAuthPolicy",
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken"
    ],
  },
```

```
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

Note

Cette stratégie contient des instructions qui autorisent l'accès à un nombre de ressources AWS potentiellement important. Pour limiter l'accès d'AWS CodeBuild à certaines ressources AWS, modifiez la valeur du tableau `Resource`. Pour plus d'informations, consultez la documentation de sécurité pour le service AWS.

6. Sur la page Review Policy (Examiner une stratégie), pour Policy Name (Nom de stratégie), saisissez un nom pour la stratégie (par exemple, **CodeBuildServiceRolePolicy**), puis choisissez Create policy (Créer une stratégie).

Note

Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.

7. Dans le panneau de navigation, choisissez Roles (Rôles).
8. Sélectionnez Create role (Créer un rôle).
9. Sur la page Créer un rôle, avec AWSService déjà sélectionné, choisissez CodeBuild, puis choisissez Next:Permissions.
10. Sur la page Joindre des politiques d'autorisation CodeBuildServiceRolePolicy, sélectionnez, puis choisissez Suivant : Révision.
11. Sur la page Create role and review (Créer un rôle et vérifier), pour Role name (Nom du rôle), saisissez le nom du rôle (par exemple, **CodeBuildServiceRole**), puis choisissez Create role (Créer un rôle).

Pour créer un rôle CodeBuild de service (AWS CLI)

1. Assurez-vous d'avoir configuré leAWS CLI avec la cléAWS d'accès et la clé d'accèsAWS secrète qui correspondent à l'une des entités IAM, comme décrit dans la procédure précédente. Pour plus d'informations, consultez [Préparation de l'installation de l'AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface.
2. Dans un répertoire vide sur le poste de travail ou l'instance local où l'AWS CLI est installée, créez deux fichiers nommés`create-role.json` et `put-role-policy.json`. Si vous choisissez des noms de fichier différents, substituez-les tout au long de cette procédure.

`create-role.json`:

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

Note

Nous vous recommandons d'utiliser les clés de condition `aws:SourceAccount` et `aws:SourceArn` pour vous protéger contre [le problème du député confus](#). Par exemple, vous pouvez modifier la politique d'approbation précédente avec les blocs de condition suivants. `aws:SourceAccount` Il s'agit du propriétaire du CodeBuild projet et `aws:SourceArn` l'ARN CodeBuild du projet.

Si vous souhaitez limiter votre rôle de service à unAWS compte, ce`create-role.json` pourrait ressembler à ceci :

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "codebuild.amazonaws.com"  
    },  
    "Action": "sts:AssumeRole",  
    "Condition": {  
      "StringEquals": {  
        "aws:SourceAccount": [  
          "account-ID"  
        ]  
      }  
    }  
  }  
]
```

Si vous souhaitez limiter votre rôle de service à un CodeBuild projet spécifique, cela crée `role.json` pourrait ressembler à ceci :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "codebuild.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"  
        }  
      }  
    }  
  ]  
}
```

Note

Si vous ne connaissez pas ou n'avez pas encore choisi le nom de votre CodeBuild projet et que vous souhaitez appliquer une politique d'approbation à un modèle ARN particulier, vous pouvez remplacer cette partie de l'ARN par un caractère générique (*). Après avoir créé votre projet, vous pouvez mettre à jour la politique de confiance.

put-role-policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    "Sid": "S3PutObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

Note

Cette stratégie contient des instructions qui autorisent l'accès à un nombre de ressources AWS potentiellement important. Pour limiter l'accès d'AWS CodeBuild à certaines ressources AWS, modifiez la valeur du tableau `Resource`. Pour plus d'informations, consultez la documentation de sécurité pour le service AWS.

3. Placez-vous dans le répertoire dans lequel vous avez enregistré les fichiers précédents, puis exécutez les deux commandes suivantes, une par une, dans cet ordre. Vous pouvez utiliser des valeurs différentes pour `CodeBuildServiceRole` et `CodeBuildServiceRolePolicy`, mais veillez à les substituer ici.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

Création et configuration d'une clé gérée par le client pour CodeBuild

AWS CodeBuild Pour crypter ses artefacts de sortie de génération, il doit accéder à une clé KMS. Par défaut, CodeBuild utilise la clé gérée par AWS pour Amazon S3 dans votre AWS compte.

Si vous ne souhaitez pas utiliser la clé gérée par AWS, vous devez créer et configurer vous-même une clé gérée par le client. Cette section décrit comment procéder avec la console IAM.

Pour plus d'informations sur les clés gérées par le client, consultez la section [AWS Key Management Service Concepts](#) et [création de clés](#) du manuel du AWS KMS développeur.

Pour configurer une clé gérée par le client à des fins d'utilisation par CodeBuild, suivez les instructions de la section « Comment modifier une politique clé » de la section [Modification d'une politique clé](#) dans le Guide du AWS KMS développeur. Ajoutez ensuite les instructions suivantes (entre **### BEGIN ADDING STATEMENTS HERE ###** et **### END ADDING STATEMENTS HERE ###**) à la stratégie de clé. Des ellipses (. . .) sont utilisées par souci de concision et pour vous aider à déterminer l'endroit où ajouter les instructions. Ne supprimez aucune instruction, et n'entrez pas ces ellipses dans la stratégie de clé.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENTS HERE ###
    {
      "Sid": "Allow access through Amazon S3 for all principals in the account that are
authorized to use Amazon S3",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```

        "kms:ViaService": "s3.region-ID.amazonaws.com",
        "kms:CallerAccount": "account-ID"
    }
}
},
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::account-ID:role/CodeBuild-service-role"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
### END ADDING STATEMENTS HERE ###
{
    "Sid": "Enable IAM User Permissions",
    ...
},
{
    "Sid": "Allow access for Key Administrators",
    ...
},
{
    "Sid": "Allow use of the key",
    ...
},
{
    "Sid": "Allow attachment of persistent resources",
    ...
}
]
}

```

- *Region-ID* représente l'ID de laAWS région dans laquelle se CodeBuild trouvent les compartiments Amazon S3 associés (par exemple,us-east-1).
- *L'ID* duAWS compte propriétaire de la clé gérée par le client.

- *CodeBuild-service-role* représente le nom du rôle de CodeBuild service que vous avez créé ou identifié plus haut dans cette rubrique.

Note

Pour créer ou configurer une clé gérée par le client via la console IAM, vous devez d'abord vous connecter à l'aide de l'AWS Management Console de l'une des méthodes suivantes :

- Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section The Account Root User](#) dans le guide de l'utilisateur.
- Un utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez [la section Création de votre premier utilisateur et groupe Compte AWS root](#) dans le Guide de l'utilisateur.
- Utilisateur de votre AWS compte autorisé à créer ou à modifier la clé gérée par le client. Pour plus d'informations, consultez [Autorisations requises pour utiliser la console AWS KMS](#) dans le Manuel du développeur AWS KMS.

Installation et configuration de l'AWS CLI

Pour y accéder AWS CodeBuild, vous pouvez utiliser la console AWS CLI avec ou à la place de la CodeBuild console, de la CodePipeline console ou AWS des SDK. Pour installer et configurer le AWS CLI, consultez la section [Getting Setup with the AWS Command Line Interface](#) dans le Guide de l'utilisateur de l'AWS Command Line Interface.

1. Exécutez la commande suivante pour vérifier si vous avez installé les AWS CLI supports CodeBuild :

```
aws codebuild list-builds
```

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

```
{  
  "ids": []  
}
```

Les crochets vides indiquent que vous n'avez pas encore exécuté de générations.

2. Si une erreur est générée, vous devez désinstaller votre version actuelle de l'AWS CLI, puis installer la dernière version. Pour plus d'informations, consultez [Désinstallation de l'AWS CLI](#) et [Installation de l'AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface.

Référence des commandes en ligne pour AWS CodeBuild

L'AWS CLI fournit des commandes pour automatiser AWS CodeBuild. Utilisez les informations de cette rubrique en complément des [AWS Command Line Interface Guide de l'utilisateur](#) et [l'AWS CLIRéférence pourAWS CodeBuild](#).

Ce n'est pas ce que vous recherchez ? Si vous souhaitez utiliser l'interface AWS SDK pour appeler CodeBuild, consultez le [Référence des kits SDK et des outils AWS](#).

Pour utiliser les informations de cette rubrique, vous devez avoir installé l'interface AWS CLI et l'a configuré pour être utilisé avec CodeBuild, comme décrit dans [Installation et configuration de l'AWS CLI](#).

Pour utiliser le plugin AWS CLI pour spécifier le point de terminaison pour CodeBuild, consultez [Spécification du point de terminaison AWS CodeBuild \(AWS CLI\)](#).

Exécutez cette commande pour obtenir la liste des commandes CodeBuild.

```
aws codebuild help
```

Exécutez cette commande pour obtenir des informations sur une commande CodeBuild, où *nom-commande* est le nom de la commande.

```
aws codebuild command-name help
```

Les commandes CodeBuild sont les suivantes :

- `batch-delete-builds`: Supprime une ou plusieurs générations dans CodeBuild. Pour plus d'informations, consultez [Suppression de générations \(AWS CLI\)](#).
- `batch-get-builds`: Obtient des informations sur plusieurs générations dans CodeBuild. Pour plus d'informations, consultez [Affichage des détails d'une génération \(AWS CLI\)](#).

- `batch-get-projects` : Obtient des informations sur un ou plusieurs projets de génération spécifiés. Pour plus d'informations, consultez [Affichage des détails d'un projet de génération \(AWS CLI\)](#).
- `create-project` : crée un projet de génération. Pour plus d'informations, consultez [Création d'un projet de génération \(AWS CLI\)](#).
- `delete-project` : supprime un projet de génération. Pour plus d'informations, consultez [Suppression d'un projet de génération \(AWS CLI\)](#).
- `list-builds`: Répertorie les Amazon Resource Names (ARN) pour des générations dans CodeBuild. Pour plus d'informations, consultez [Affichage d'une liste d'ID de génération \(AWS CLI\)](#).
- `list-builds-for-project` : Obtient une liste d'ID de génération associés à un projet de génération spécifié. Pour plus d'informations, consultez [Affichage d'une liste d'ID de génération pour un projet de génération \(AWS CLI\)](#).
- `list-curated-environment-images`: Obtient une liste d'images Docker gérées par CodeBuild que vous pouvez utiliser pour vos générations. Pour plus d'informations, consultez [Images Docker fournies par CodeBuild](#).
- `list-projects` : Obtient une liste de noms de projet de génération. Pour plus d'informations, consultez [Affichage d'une liste de noms de projet de génération \(AWS CLI\)](#).
- `start-build` : démarre l'exécution d'une génération. Pour plus d'informations, consultez [Exécution d'une génération \(AWS CLI\)](#).
- `stop-build` : Tente d'arrêter l'exécution de la génération spécifiée. Pour plus d'informations, consultez [Arrêt d'une génération \(AWS CLI\)](#).
- `update-project` : Modifie des informations sur le projet de génération spécifié. Pour de plus amples informations, veuillez consulter [Modification des paramètres d'un projet de génération \(AWS CLI\)](#).

Référence des kits SDK et des outils AWS pour AWS CodeBuild

Pour utiliser l'un des kits SDK ou des outils AWS pour automatiser AWS CodeBuild, consultez les ressources suivantes.

Si vous souhaitez utiliser le AWS CLI pour exécuter CodeBuild, consultez le [Référence des commandes en ligne](#).

Kits SDK et outils AWS pris en charge pour AWS CodeBuild

Procédez comme suit :AWS Les kits SDK et outils prennent en charge CodeBuild :

- Le [kit SDK AWS pour C++](#). Pour plus d'informations, consultez la section sur l'espace de noms [Aws::CodeBuild](#) du document AWS SDK for C++ API Reference.
- Le [kit SDK AWS pour Go](#). Pour plus d'informations, consultez le [.codebuild](#)Section duAWS Référence d'API du kit SDK for Go.
- Le [kit SDK AWS pour Java](#). Pour de plus amples informations, veuillez consulter les sections `com.amazonaws.services.codebuild` et `com.amazonaws.services.codebuild.model` du document [Référence d'API du kit AWS SDK pour Java](#).
- Le [kit AWS SDK pour JavaScript dans le navigateur](#) et le [kit AWS SDK pour JavaScript dans Node.js](#). Pour plus d'informations, consultez le [.Classe : AWS.CodeBuild](#)Section duAWS Référence d'API du kit SDK for JavaScript.
- Le [kit SDK AWS pour .NET](#). Pour plus d'informations, consultez le [.Construire Amazon.codeetAmazon.CodeBuild.Model](#)sections d'espace de noms duAWS Référence d'API du kit SDK for .NET.
- Le [kit SDK AWS pour PHP](#). Pour plus d'informations, consultez le [.Espace de noms Aws \ CodeBuild](#)Section duAWS Référence d'API du kit SDK for PHP.
- Le [kit SDK AWS pour Python \(Boto3\)](#). Pour plus d'informations, consultez le [.CodeBuild](#)Section duDocumentation du Boto 3.
- Le [kit SDK AWS pour Ruby](#). Pour plus d'informations, consultez le [.Module : Aws:: CodeBuild](#)Section duAWS Référence d'API du kit SDK for Ruby.
- Les [outils AWS pour PowerShell](#). Pour plus d'informations, consultez la section [AWS CodeBuild](#) du document AWS Tools for PowerShell Cmdlet Reference.

Spécification du point de terminaison AWS CodeBuild

Vous pouvez utiliser l'AWS Command Line Interface (AWS CLI) ou un des kits SDK AWS pour spécifier le point de terminaison utilisé par AWS CodeBuild. Il existe un point de terminaison pour chaque région dans laquelle CodeBuild est disponible. En plus d'un point de terminaison régional, quatre régions ont également un point de terminaison FIPS (Federal Information Processing Standards). Pour de plus amples informations sur les points de terminaison FIPS, veuillez consulter [Présentation de FIPS 140-2](#).

La spécification d'un point de terminaison est facultative. Si vous n'indiquez pas explicitement à CodeBuild le point de terminaison à utiliser, le service utilise le point de terminaison associé à la région que votre compte AWS utilise. CodeBuild n'utilise jamais par défaut un point de terminaison FIPS. Si vous souhaitez utiliser un point de terminaison FIPS, vous devez lui associer CodeBuild via une des méthodes suivantes.

Note

Vous pouvez utiliser un alias ou un nom de région pour spécifier un point de terminaison avec un kit de développement logiciel (SDK) AWS. Si vous utilisez l'AWS CLI, vous devez utiliser le nom de point de terminaison complet.

Pour connaître les points de terminaison pouvant être utilisés avec CodeBuild, veuillez consulter [Régions et points de terminaison CodeBuild](#).

Rubriques

- [Spécification du point de terminaison AWS CodeBuild \(AWS CLI\)](#)
- [Spécification du point de terminaison AWS CodeBuild \(kit SDK AWS\)](#)

Spécification du point de terminaison AWS CodeBuild (AWS CLI)

Vous pouvez utiliser l'interface AWS CLI pour spécifier le point de terminaison via lequel AWS CodeBuild est accédé par l'intermédiaire de l'argument `--endpoint-url` dans n'importe quelle commande CodeBuild. Par exemple, exécutez cette commande pour obtenir la liste des noms de version du projet à l'aide du point de terminaison FIPS (Federal Information Processing Standards) dans la région USA Est (Virginie du Nord) :

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Incluez `https://` au début du point de terminaison.

L'argument AWS CLI `--endpoint-url` est disponible pour tous les services AWS. Pour plus d'informations sur cet AWS CLI argument et sur d'autres, consultez [AWS CLI Command Reference](#).

Spécification du point de terminaison AWS CodeBuild (kit SDK AWS)

Vous pouvez utiliser un kit de développement logiciel (SDK) AWS pour spécifier le point de terminaison via lequel AWS CodeBuild est accédé. Bien que cet exemple utilise le [kit de développement logiciel \(SDK\) AWS pour Java](#), vous pouvez spécifier le point de terminaison avec les autres kits SDK AWS.

Utilisez la `withEndpointConfiguration` méthode lors de la création du `AWSCodeBuild` client. Voici le format à utiliser :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
    "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Pour plus d'informations sur `AWSCodeBuildClientBuilder`, voir [ClassAWSCodeBuildClientBuilder](#).

Les informations d'identification utilisées dans `withCredentials` doivent être de type `AWSCredentialsProvider`. Pour de plus amples informations, veuillez consulter [Utilisation des informations d'identification AWS](#).

N'incluez pas `https://` au début du point de terminaison.

Si vous souhaitez spécifier un point de terminaison non conforme à la norme FIPS, vous pouvez utiliser la région plutôt que le point de terminaison. Par exemple, pour spécifier le point de terminaison dans la région USA Est (Virginie du Nord), vous pouvez utiliser `us-east-1` au lieu du nom complet du point de terminaison, `codebuild.us-east-1.amazonaws.com`.

Si vous souhaitez spécifier un point de terminaison FIPS, vous pouvez utiliser un alias pour simplifier votre code. Seuls les points de terminaison FIPS ont un alias. Les autres points de terminaison doivent être spécifiés à l'aide de leur nom de région ou de leur nom complet.

Le tableau suivant répertorie les alias pour chacun des quatre points de terminaison FIPS disponibles :

| Nom de la région | Région | Point de terminaison | Alias |
|--------------------------------|-----------|--|----------------|
| US East (Virginie du Nord) | us-east-1 | codebuild-fips.us-east-1.amazonaws.com | us-east-1-fips |
| USA Est (Ohio) | us-east-2 | codebuild-fips.us-east-2.amazonaws.com | us-east-2-fips |
| USA Ouest (Californie du Nord) | us-west-1 | codebuild-fips.us-west-1.amazonaws.com | us-west-1-fips |
| USA Ouest (Oregon) | us-west-2 | codebuild-fips.us-west-2.amazonaws.com | us-west-2-fips |

Pour spécifier l'utilisation du point de terminaison FIPS dans la région USA Ouest (Oregon) à l'aide d'un alias :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Pour spécifier l'utilisation du point de terminaison non FIPS dans la région USA Est (Virginie du Nord) :

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
"us-east-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Pour spécifier l'utilisation du point de terminaison non FIPS dans la région Asie-Pacifique (Mumbai) :

```

AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
"ap-south-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();

```

Utilisation d'AWS CodePipeline avec AWS CodeBuild pour tester le code et exécuter des générations

Vous pouvez automatiser le processus de lancement en utilisant AWS CodePipeline pour tester le code et exécuter les générations avec AWS CodeBuild.

Le tableau suivant répertorie les tâches et les méthodes disponibles pour ces tâches. L'utilisation des kits SDK AWS pour effectuer ces tâches n'entre pas dans le cadre de cette rubrique.

| Tâche | Approches disponibles | Approches décrites dans cette rubrique |
|--|---|--|
| Créez un pipeline de livraison continue (CD) CodePipeline qui automatise les builds avec CodeBuild | <ul style="list-style-type: none"> CodePipeline console AWS CLI Kits SDK AWS | <ul style="list-style-type: none"> Utilisation de la console CodePipeline Utilisation de l'AWS CLI Vous pouvez adapter les informations de cette rubrique pour utiliser les kits SDK AWS. Pour plus d'informations, consultez la documentation relative aux <code>create-pipeline</code> actions correspondant à votre langage de programmation dans la section SDK de Tools for Amazon Web Services ou CreatePipeline dans le Guide de référence des AWS CodePipeline API. |
| Ajoutez l'automatisation des tests et du CodeBuild développement à un pipeline | <ul style="list-style-type: none"> CodePipeline console AWS CLI Kits SDK AWS | <ul style="list-style-type: none"> Utiliser la CodePipeline console pour ajouter l'automatisation des builds Utiliser la CodePipeline console pour ajouter l'automatisation des tests |

| Tâche | Approches disponibles | Approches décrites dans cette rubrique |
|----------------------------|-----------------------|--|
| existant dans CodePipeline | | <ul style="list-style-type: none">• Pour leAWS CLI, vous pouvez adapter les informations de cette rubrique afin de créer un pipeline contenant une action de CodeBuild création ou une action de test. Pour plus d'informations, voir Modifier un pipeline (AWS CLI) et la référence de structure du CodePipeline pipeline dans le Guide de AWS CodePipeline l'utilisa teur.• Vous pouvez adapter les informations de cette rubrique pour utiliser les kits SDK AWS. Pour plus d'informa tions, consultez la documentation des update-pi peline actions correspondant à votre langage de programmation dans la section SDK de Tools for Amazon Web Services ou UpdatePipeline dans la référence des AWS CodePipeline API. |

Prérequis

1. Répondez aux questions de [Planification d'une génération](#).
2. Si vous utilisez un utilisateur pour accéder CodePipeline au lieu d'un compte AWS root ou d'un utilisateur administrateur, associez la politique gérée nommée `AWSCodePipelineFullAccess` à l'utilisateur (ou au groupe IAM auquel appartient l'utilisateur). L'utilisation d'un compte racine AWS n'est pas recommandée. Cette stratégie accorde à l'utilisateur l'autorisation de créer le pipeline dans CodePipeline. Pour plus d'informations, consultez la section [Joindre des politiques gérées](#) dans le guide de l'utilisateur.

Note

L'entité IAM qui attache la politique à l'utilisateur (ou au groupe IAM auquel appartient l'utilisateur) doit être autorisée dans IAM à attacher des politiques. Pour plus d'informations, consultez la section [Délégation d'autorisations pour administrer les utilisateurs, les groupes et les informations d'identification IAM](#) dans le guide de l'utilisateur.

3. Créez un rôle de CodePipeline service, si aucun rôle n'est déjà disponible dans votre AWS compte. CodePipeline utilise ce rôle de service pour interagir avec d'autres AWS services AWS CodeBuild, y compris en votre nom. Par exemple, pour utiliser le AWS CLI pour créer un rôle CodePipeline de service, exécutez la `create-role` commande IAM :

Pour Linux, macOS ou Unix :

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role
--assume-role-policy-document '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Principal":
{"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}'}
```

Pour Windows :

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-
role-policy-document '{"Version":"2012-10-17","Statement":{"Effect":
"Allow","Principal":{"Service":"codepipeline.amazonaws.com"},"Action":
"sts:AssumeRole"}'}
```

Note

L'entité IAM qui crée ce rôle de CodePipeline service doit être autorisée dans IAM à créer des rôles de service.

4. Après avoir créé un rôle de CodePipeline service ou identifié un rôle existant, vous devez ajouter la politique de rôle de CodePipeline service par défaut au rôle de service, comme décrit dans la section [Révision de la politique de rôle de CodePipeline service par défaut](#) dans le Guide de AWS CodePipeline l'utilisateur, si elle ne fait pas déjà partie de la politique du rôle.

Note

L'entité IAM qui ajoute cette politique de rôle de CodePipeline service doit être autorisée dans IAM à ajouter des politiques de rôle de service aux rôles de service.

5. Créez et chargez le code source dans un type de référentiel pris en charge par CodeBuild et CodePipeline CodeCommit, tel qu'Amazon S3, Bitbucket ou GitHub. Le code source doit contenir un fichier de spécification de génération, mais vous pourrez en déclarer un lorsque vous définirez

un projet de génération ultérieurement dans cette rubrique. Pour plus d'informations, consultez le [Référence des spécifications de génération](#).

Important

Si vous prévoyez d'utiliser le pipeline pour déployer le code source généré, l'artefact de sortie de génération doit être compatible avec le système de déploiement que vous utilisez.

- Pour AWS OpsWorks, voir [Source de l'application](#) et [Utilisation CodePipeline avec AWS OpsWorks](#) dans le guide de AWS OpsWorks l'utilisateur.

Rubriques

- [Création d'un pipeline utilisant CodeBuild \(console CodePipeline\)](#)
- [Création d'un pipeline utilisant CodeBuild \(AWS CLI\)](#)
- [Ajout d'une action de génération CodeBuild à un pipeline \(console CodePipeline\)](#)
- [Ajout d'une action de test CodeBuild à un pipeline \(console CodePipeline\)](#)

Création d'un pipeline utilisant CodeBuild (console CodePipeline)

Utilisez la procédure suivante pour créer un pipeline qui utilise CodeBuild pour générer et déployer votre code source.

Pour créer un pipeline qui teste uniquement votre code source :

- Utilisez la procédure suivante pour créer le pipeline, puis supprimez les étapes de génération et bêta du pipeline. Utilisez ensuite la procédure [Ajout d'une action de test CodeBuild à un pipeline \(console CodePipeline\)](#) décrite dans cette rubrique pour ajouter au pipeline une action de test qui utilise CodeBuild.
- Utilisez l'une des autres procédures de cette rubrique pour créer le pipeline, puis utilisez la procédure [Ajout d'une action de test CodeBuild à un pipeline \(console CodePipeline\)](#) décrite dans cette rubrique pour ajouter au pipeline une action de test qui utilise CodeBuild.

Pour utiliser l'assistant de création de pipeline dans CodePipeline afin de créer un pipeline qui utilise CodeBuild

1. Connectez-vous à AWS Management Console en utilisant :

- Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le Guide de l'utilisateur.
- Un utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur AWS root et de votre premier groupe](#) dans le Guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à utiliser l'ensemble minimum d'actions suivant :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Ouvrez la AWS CodePipeline console à l'[adresse https://console.aws.amazon.com/codesuite/codepipeline/home](https://console.aws.amazon.com/codesuite/codepipeline/home).
3. Dans le sélecteur de AWS région, choisissez la AWS région dans laquelle se trouvent les AWS ressources de votre projet de construction. Il doit s'agir d'une AWS région prise en charge par CodeBuild. Pour de plus amples informations, veuillez consulter [AWS CodeBuild](#) dans le Référence générale d'Amazon Web Services.

4. Créez un pipeline. Si une page d'information CodePipeline s'affiche, choisissez Créer un pipeline. Si une page Tous les pipelines s'affiche, choisissez Créer un pipeline.
5. Sur la page Étape 1 : Choisir des paramètres de pipeline, pour Nom du pipeline, saisissez un nom pour le pipeline, par exemple, **CodeBuildDemoPipeline**. Si vous choisissez un autre nom, veillez à l'utiliser tout au long de cette procédure.
6. Pour Nom du rôle, sélectionnez l'une des options suivantes :

Choisissez New service role (Nouveau rôle de service) et dans Nom du rôle, saisissez le nom de votre nouveau rôle de service.

Choisissez Existing service role (Rôle de service existant), puis choisissez le rôle de service CodePipeline que vous avez créé ou identifié dans le cadre des prérequis de cette rubrique.

7. Pour Artifact store (Magasin d'artefacts), effectuez l'une des opérations suivantes :
 - Choisissez l'emplacement par défaut pour utiliser le magasin d'artefacts par défaut, tel que le compartiment d'artefacts S3 désigné comme emplacement par défaut, pour votre pipeline dans laAWS région que vous avez sélectionnée pour votre pipeline.
 - Choisissez Emplacement personnalisé si vous avez déjà créé un magasin d'artefacts, tel qu'un compartiment d'artefacts S3, dans la mêmeAWS région que votre pipeline.

Note

Il ne s'agit pas du compartiment source du code source pour votre pipeline. Il s'agit du magasin d'artefacts pour votre pipeline. Un magasin d'artefacts distinct, tel qu'un compartiment S3, est nécessaire pour chaque pipeline, estAWS nécessaire pour chaque pipeline.

8. Choisissez Suivant.
9. Sur la page Étape 2 : Ajouter une étape source, pour Fournisseur de source, effectuez l'une des actions suivantes :
 - Si votre code source est stocké dans un compartiment S3, choisissez Amazon S3. Pour Bucket (Compartiment), sélectionnez le compartiment S3 qui contient votre code source. Pour S3 object key (Clé d'objet S3), saisissez le nom du fichier qui contient le code source (par exemple *file-name*.zip). Choisissez Suivant.

- Si votre code source est stocké dans un compartiment AWS CodeCommit, choisissez CodeCommit. Pour Repository name, choisissez le nom du référentiel qui contient le code source. Pour Branch name (Nom de branche), choisissez le nom de la branche qui représente la version du code source à générer. Choisissez Suivant.
- Si votre code source est stocké dans un GitHub référentiel, choisissez GitHub. Choisissez Connect à GitHub et suivez les instructions pour vous authentifier auprès de GitHub. Pour Repository, choisissez le nom du référentiel qui contient le code source. Pour Branch (Branche), choisissez le nom de la branche qui représente la version du code source à générer.

Choisissez Suivant.

10. Sur la page Étape 3: Ajouter une étape de génération, pour Fournisseur de génération, choisissez CodeBuild.
11. Si vous avez déjà un projet de génération que vous souhaitez utiliser, dans Nom du projet, choisissez le nom du projet de génération et passez à l'étape suivante de cette procédure.

Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

Si vous choisissez un projet de génération existant, celui-ci doit avoir des paramètres d'artefact de sortie déjà définis (même si CodePipeline les remplace). Pour plus d'informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).

Important


Si vous activez des webhooks pour un projet CodeBuild et que le projet est utilisé en tant qu'étape de génération dans CodePipeline, deux générations identiques sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la console AWS CodeBuild, décochez la case Webhook. Pour plus d'informations, veuillez consulter [Modification des paramètres d'un projet de génération \(console\)](#).

12. Sur la page Étape 4 : Ajouter une étape de déploiement, effectuez l'une des opérations suivantes :

- Si vous ne souhaitez pas déployer l'artefact de sortie de génération, choisissez Ignorer et confirmez ce choix lorsque vous y êtes invité.
- Si vous souhaitez déployer l'artefact de sortie de génération, pour Deploy provider (Fournisseur de déploiement), choisissez un fournisseur de déploiement, puis spécifiez les paramètres lorsque vous y êtes invité.

Choisissez Suivant.

13. Sur la page Vérification, vérifiez vos sélections, puis choisissez Créer un pipeline.
14. Une fois que le pipeline s'exécute correctement, vous pouvez obtenir l'artefact de sortie de la génération. Avec le pipeline affiché dans la console CodePipeline, dans l'action Génération, sélectionnez l'info-bulle. Notez la valeur de Output artefact (par exemple, MyAppBuild).

 Note

Vous pouvez également obtenir l'artefact de sortie de la génération en choisissant le lien Artefacts de génération sur la page des détails de génération dans la console CodeBuild. Pour accéder à cette page, ignorez les autres étapes de cette procédure et consultez [Affichage des détails d'une génération \(console\)](#).

15. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
16. Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format `codepipeline-region-ID-random-number`. Vous pouvez utiliser leAWS CLI pour exécuter laCodePipelineget-pipeline commande afin d'obtenir le nom du compartiment, où se `my-pipeline-name`trouve le nom d'affichage de votre pipeline :

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dans la sortie, l'objet pipeline contient un objet artifactStore contenant une valeur location avec le nom du compartiment.

17. Ouvrez le dossier correspondant au nom de votre pipeline (selon la longueur du nom du pipeline, le nom de dossier peut être tronqué), puis ouvrez le dossier correspondant à la valeur pour Output artifact (Artefact de sortie) que vous avez notée précédemment.
18. Extrayez le contenu du fichier . Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage Last Modified. (Vous pouvez attribuer au fichier l'extension

.zip pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.

19. Si vous avez demandé à CodePipeline de déployer l'artefact de sortie de génération, utilisez les instructions du fournisseur de déploiement pour obtenir l'artefact de sortie de génération sur les cibles de déploiement.

Création d'un pipeline utilisant CodeBuild (AWS CLI)

Utilisez la procédure suivante pour créer un pipeline qui utilise CodeBuild pour générer votre code source.

AWS CLI Pour créer un pipeline qui déploie le code source que vous avez créé ou qui teste uniquement votre code source, vous pouvez adapter les instructions de la section [Modifier un pipeline \(AWS CLI\)](#) et la [référence à la structure du CodePipeline pipeline](#) dans le Guide de l'utilisateur AWS CodePipeline.

1. Créez ou identifiez un projet de génération dans CodeBuild. Pour plus d'informations, veuillez consulter [Création d'un projet de génération](#).

Important

Le projet de génération doit définir des paramètres d'artefact de sortie de génération (même si CodePipeline va remplacer). Pour de plus amples informations, consultez la description de artifacts dans [Création d'un projet de génération \(AWS CLI\)](#).

2. Assurez-vous d'avoir configuré le AWS CLI avec la clé AWS d'accès et la clé d'accès AWS secrète qui correspondent à l'une des entités IAM décrites dans cette rubrique. Pour de plus amples informations, veuillez consulter [Préparation de l'installation de l'AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface.
3. Créez un fichier au format JSON représentant la structure du pipeline. Nommez le fichier `create-pipeline.json` ou avec un nom similaire. Par exemple, cette structure au format JSON crée un pipeline avec une action source qui référence un compartiment d'entrée S3 et une action de génération utilisant CodeBuild :

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-name>",
```

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

```
        "name": "default"
      }
    ],
    "configuration": {
      "ProjectName": "<build-project-name>"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<CodePipeline-internal-bucket-name>"
},
"name": "<my-pipeline-name>",
"version": 1
}
}
```

Dans ces données au format JSON :

- La valeur de `roleArn` doit correspondre à l'ARN du rôle de service CodePipeline que vous avez créé ou identifié dans le cadre des prérequis.
- Les valeurs de `S3Bucket` et `S3ObjectKey` dans `configuration` supposent que le code source est stocké dans un compartiment S3. Pour les paramètres pour d'autres types de référentiel de code source, veuillez consulter [Référence sur la structure du pipeline CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline.
- La valeur de `ProjectName` est le nom du projet de génération CodeBuild créé précédemment dans cette procédure.
- La valeur de `location` est le nom du compartiment S3 utilisé par ce pipeline. Pour de plus amples informations, veuillez consulter [Création d'une stratégie pour un compartiment S3 en vue de l'utiliser comme magasin d'artefacts pour CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline.
- La valeur de `name` est le nom de ce pipeline. Tous les noms de pipeline doivent être uniques dans votre compte.

Même si ces données décrivent uniquement une action source et une action de génération, vous pouvez ajouter des actions pour les activités liées au test, au déploiement de l'artefact de sortie

de génération, à l'appel des fonctions AWS Lambda, etc. Pour de plus amples informations, veuillez consulter [Référence sur la structure du pipeline AWS CodePipeline](#) dans le Guide de l'utilisateur AWS CodePipeline.

4. Placez-vous dans le dossier qui contient le fichier JSON, puis exécutez la commande CodePipeline [create-pipeline](#) en spécifiant le nom de fichier :

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

Note

Vous devez créer le pipeline dans une région AWS qui prend en charge CodeBuild. Pour de plus amples informations, veuillez consulter [AWS CodeBuild](#) dans la Référence générale d'Amazon Web Services.

Les données au format JSON s'affichent dans la sortie, et CodePipeline crée le pipeline.

5. Pour obtenir des informations sur l'état du pipeline, exécutez la commande CodePipeline [get-pipeline-state](#) en spécifiant le nom du pipeline :

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Dans la sortie, recherchez les informations qui confirment que la génération a réussi. Des ellipses (...) sont utilisées pour les données qui ont été omises par souci de concision.

```
{
  ...
  "stageStates": [
    ...
    {
      "actionStates": [
        {
          "actionName": "CodeBuild",
          "latestExecution": {
            "status": "SUCCEEDED",
            ...
          },
          ...
        }
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

Si vous exécutez cette commande trop tôt, vous risquez de ne voir aucune information sur l'action de génération. Vous devrez peut-être exécuter cette commande plusieurs fois jusqu'à ce que le pipeline ait fini d'exécuter l'action de génération.

- Après une génération réussie, suivez ces instructions pour obtenir l'artefact de sortie de la génération. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.

Note

Vous pouvez également obtenir l'artefact de sortie de la génération en choisissant le lien Artefacts de génération sur la page des détails de génération associées dans la console CodeBuild. Pour accéder à cette page, ignorez les autres étapes de cette procédure et consultez [Affichage des détails d'une génération \(console\)](#).

- Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format `codepipeline-<region-ID>-<random-number>`. Vous pouvez obtenir le nom du compartiment à partir du fichier `create-pipeline.json` ou en exécutant la commande `CodePipeline get-pipeline`.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

Dans la sortie, l'objet `pipeline` contient un objet `artifactStore` contenant une valeur `location` avec le nom du compartiment.

- Ouvrez le dossier correspondant au nom de votre pipeline (par exemple, *<pipeline-name>*).
- Dans ce dossier, ouvrez le dossier nommé `default` (par défaut).
- Extrayez le contenu du fichier `.`. Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage `Last Modified`. (Vous pouvez attribuer au fichier une extension `.zip` pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.

Ajout d'une action de génération CodeBuild à un pipeline (console CodePipeline)

1. Connectez-vous à AWS Management Console en utilisant :
 - Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le Guide de l'utilisateur.
 - Un utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la [section Création de votre premier utilisateur Compte AWS root et de votre premier groupe](#) dans le Guide de l'utilisateur.
 - Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Ouvrez la CodePipeline console à l'[adresse https://console.aws.amazon.com/codesuite/codepipeline/home](https://console.aws.amazon.com/codesuite/codepipeline/home).
3. Dans le sélecteur de AWS région, choisissez la AWS région dans laquelle se trouve votre pipeline. Il doit s'agir d'une région dans laquelle CodeBuild est pris en charge. Pour de plus

amples informations, veuillez consulter [CodeBuild](#) dans le Référence générale d'Amazon Web Services.

4. Sur la page Tous les pipelines, choisissez le nom du pipeline.
5. Sur la page des détails du pipeline, dans l'action Source, sélectionnez l'info-bulle. Notez la valeur de Output artefact (par exemple, MyApp).

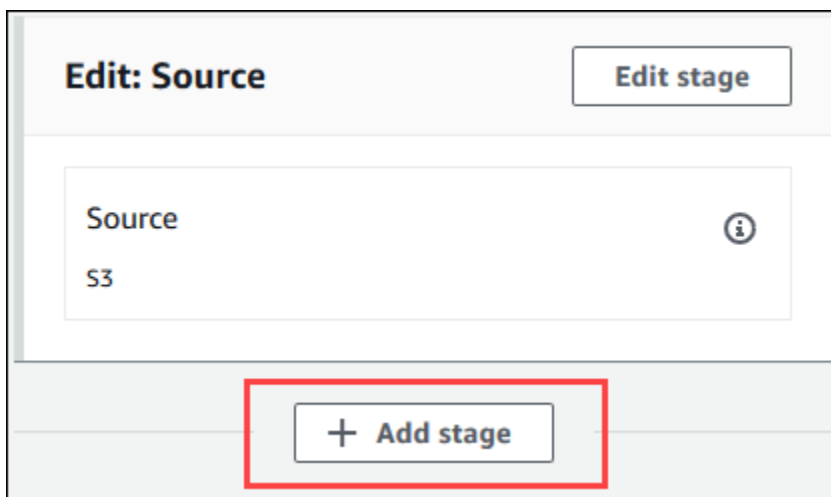
Note

Cette procédure vous explique comment ajouter une action de génération lors d'une étape de génération, entre les étapes Source et Bêta. Si vous souhaitez ajouter l'action de génération ailleurs, sélectionnez l'info-bulle de l'action juste avant l'emplacement où vous voulez ajouter l'action de génération, et notez la valeur du champ Artefact de sortie.


6. Choisissez Edit (Modifier).
7. Entre les étapes Source et Bêta, choisissez Ajouter une étape.

Note

Cette procédure vous explique comment ajouter une étape de génération entre les étapes Source et Bêta du pipeline. Pour ajouter une action de génération à une étape existante, choisissez Edit stage (Modifier une étape) à l'étape, puis passez directement à l'étape 8 de cette procédure. Pour ajouter ailleurs l'étape de génération, choisissez Add stage (Ajouter une étape) à l'emplacement voulu.



8. Pour Stage name (Nom de l'étape), saisissez le nom de l'étape de génération (par exemple, **Build**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
9. Au sein de l'étape sélectionnée, choisissez Ajouter une action.

 Note

Cette procédure vous explique comment ajouter l'action de génération lors d'une étape de génération. Pour ajouter ailleurs l'action de génération, choisissez Ajouter une action à l'emplacement voulu. Vous devrez peut-être d'abord choisir l'étape Modifier dans l'étape existante dans laquelle vous voulez générer l'action.

10. Dans Modifier l'action, pour Nom de l'action, saisissez un nom pour l'action (par exemple, **CodeBuild**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
11. Pour Action provider (Fournisseur d'action), choisissez CodeBuild.
12. Si vous avez déjà un projet de génération que vous souhaitez utiliser, dans Nom du projet, choisissez le nom du projet de génération et passez à l'étape suivante de cette procédure.


Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

Si vous choisissez un projet de génération existant, celui-ci doit avoir des paramètres d'artefact de sortie déjà définis (même si CodePipeline les remplace). Pour de plus amples informations, consultez la description d'Artefacts dans [Création d'un projet de génération \(console\)](#) ou [Modification des paramètres d'un projet de génération \(console\)](#).

 Important

Si vous activez des webhooks pour un projet CodeBuild et que le projet est utilisé en tant qu'étape de génération dans CodePipeline, deux générations identiques sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la console CodeBuild, décochez la case Webhook. Pour de plus amples informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).

13. Pour Input artifacts (Artefacts d'entrée), sélectionnez l'artefact de sortie que vous avez notée précédemment dans cette procédure.
14. Pour Artefacts de sortie, saisissez un nom pour l'artefact de sortie (par exemple, **MyAppBuild**).
15. Choisissez Add action.
16. Choisissez Save (Enregistrer), puis Save (Enregistrer) pour enregistrer les modifications apportées au pipeline.
17. Choisissez Release Change.
18. Une fois que le pipeline s'exécute correctement, vous pouvez obtenir l'artefact de sortie de la génération. Avec le pipeline affiché dans la console CodePipeline, dans l'action Génération, sélectionnez l'info-bulle. Notez la valeur de Output artefact (par exemple, MyAppBuild).

 Note

Vous pouvez également obtenir l'artefact de sortie de la génération en choisissant le lien Artefacts de génération sur la page des détails de génération dans la console CodeBuild. Pour accéder à cette page, consultez [Affichage des détails d'une génération \(console\)](#), puis passez directement à l'étape 31 de cette procédure.

19. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
20. Dans la liste des compartiments, ouvrez le compartiment utilisé par le pipeline. Le nom du compartiment doit suivre le format `codepipeline-region-ID-random-number`. Vous pouvez utiliser l'interface AWS CLI afin d'exécuter la commande CodePipeline `get-pipeline` pour obtenir le nom du compartiment :

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dans la sortie, l'objet `pipeline` contient un objet `artifactStore` contenant une valeur `location` avec le nom du compartiment.

21. Ouvrez le dossier correspondant au nom de votre pipeline (selon la longueur du nom du pipeline, le nom de dossier peut être tronqué), puis ouvrez le dossier correspondant à la valeur pour Output artifact (Artefact de sortie) que vous avez notée précédemment dans cette procédure.
22. Extrayez le contenu du fichier . Si ce dossier contient plusieurs fichiers, extrayez le contenu du fichier avec le dernier horodatage Last Modified. (Vous pouvez attribuer au fichier l'extension `.zip` pour pouvoir l'utiliser avec l'utilitaire ZIP de votre système.) L'artefact de sortie de génération figure dans le contenu extrait du fichier.

23. Si vous avez demandé à CodePipeline de déployer l'artefact de sortie de génération, utilisez les instructions du fournisseur de déploiement pour obtenir l'artefact de sortie de génération sur les cibles de déploiement.

Ajout d'une action de test CodeBuild à un pipeline (console CodePipeline)


1. Connectez-vous à AWS Management Console en utilisant :

- Votre compte racine AWS. Ceci n'est pas recommandé. Pour plus d'informations, consultez [la section L'utilisateur root du compte](#) dans le Guide de l'utilisateur.
- Un utilisateur administrateur de votre AWS compte. Pour plus d'informations, consultez la section [Création de votre premier utilisateur Compte AWS root et de votre premier groupe](#) dans le Guide de l'utilisateur.
- Un utilisateur de votre AWS compte autorisé à effectuer les actions minimales suivantes :

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```


2. Ouvrez la CodePipeline console à l'[adresse https://console.aws.amazon.com/codesuite/codepipeline/home](https://console.aws.amazon.com/codesuite/codepipeline/home).

3. Dans le sélecteur deAWS région, choisissez laAWS région dans laquelle se trouve votre pipeline. Il doit s'agir d'uneAWS région priseCodeBuild en charge. Pour de plus amples informations, veuillez consulter [AWS CodeBuild](#) dans le Référence générale d'Amazon Web Services.
4. Sur la page Tous les pipelines, choisissez le nom du pipeline.
5. Sur la page des détails du pipeline, dans l'action Source, sélectionnez l'info-bulle. Notez la valeur de Output artefact (par exemple, MyApp).

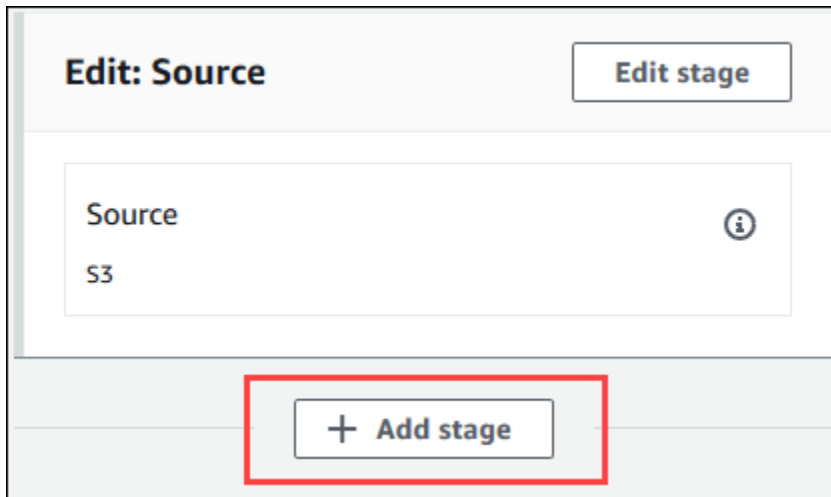
 Note

Cette procédure vous explique comment ajouter une action de test lors d'une étape de test, entre les étapes Source et Bêta. Si vous souhaitez ajouter l'action de test ailleurs, placez le pointeur de la souris sur l'action juste avant et notez la valeur pour Artefact de sortie.

6. Choisissez Edit (Modifier).
7. Immédiatement après l'étape Source, choisissez Add stage (Ajouter une étape).

 Note

Cette procédure vous explique comment ajouter au pipeline une étape de test immédiatement après l'étape Source. Pour ajouter une action de test à une étape existante, choisissez Edit stage (Modifier une étape) à l'étape, puis passez directement à l'étape 8 de cette procédure. Pour ajouter ailleurs l'étape de test, choisissez Add stage (Ajouter une étape) à l'emplacement voulu.



8. Pour Stage name (Nom de l'étape), tapez le nom de l'étape de test (par exemple, **Test**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
9. À l'étape sélectionnez, choisissez Ajouter une action.

Note

Cette procédure vous explique comment ajouter l'action de test à une étape de test. Pour ajouter l'action de test ailleurs, choisissez Add action (Ajouter une action) à l'endroit voulu. Vous devrez peut-être d'abord choisir Edit (Modifier) dans l'étape existante à laquelle vous voulez ajouter l'action de test.

10. Dans Modifier l'action, pour Nom de l'action, saisissez un nom pour l'action (par exemple, **Test**). Si vous choisissez un autre nom, utilisez-le tout au long de cette procédure.
11. Pour Action provider (Fournisseur d'actions), sous Test, choisissez CodeBuild.
12. Si vous avez déjà un projet de génération à utiliser, dans Nom du projet, choisissez le nom du projet de génération et passez à l'étape suivante de cette procédure.

Si vous devez créer un nouveau projet de CodeBuild construction, suivez les instructions fournies [Création d'un projet de génération \(console\)](#) et revenez à cette procédure.

Important

Si vous activez des webhooks pour un projet CodeBuild et que le projet est utilisé en tant qu'étape de génération dans CodePipeline, deux générations identiques

sont créées pour chaque validation. L'une est déclenchée via des webhooks et l'autre via CodePipeline. Comme la facturation est effectuée par génération, les deux générations vous sont facturées. Par conséquent, si vous utilisez CodePipeline, nous vous recommandons de désactiver les webhooks dans CodeBuild. Dans la console CodeBuild, décochez la case Webhook. Pour de plus amples informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).

13. Pour Input artifacts (Artefacts d'entrée), sélectionnez la valeur Output artifact (Artefact de sortie) que vous avez notée précédemment dans cette procédure.
14. (Facultatif) Si vous voulez que votre action de test génère un artefact de sortie et que vous avez configuré votre spécification de génération en conséquence, pour Artefact de sortie, entrez la valeur que vous voulez affecter à l'artefact de sortie.
15. Choisissez Save (Enregistrer).
16. Choisissez Release Change.
17. Une fois que le pipeline s'est exécuté avec succès, vous pouvez obtenir les résultats de test. Au cours de l'étape Test du pipeline, cliquez sur le lien hypertexte CodeBuild pour ouvrir la page de projet de génération associée dans la console CodeBuild.
18. Sur la page de projet de génération, dans la zone Historique de génération, choisissez le lien hypertexte Exécution de génération.
19. Sur la page d'exécution de la génération, dans Build logs, choisissez le lien hypertexte Afficher l'intégralité du journal pour ouvrir le journal de génération dans la CloudWatch console Amazon.
20. Faites défiler le journal de génération pour consulter les résultats de test.

Utilisation d'AWS CodeBuild avec Jenkins

Vous pouvez utiliser le plugin Jenkins pour AWS CodeBuild l'intégrer CodeBuild à vos tâches de construction Jenkins. Au lieu d'envoyer les tâches de génération aux nœuds de génération Jenkins, vous utilisez le plug-in pour envoyer vos tâches de génération à CodeBuild. Vous n'avez ainsi plus besoin d'allouer, de configurer et de gérer des nœuds de génération Jenkins.

Configuration de Jenkins

Pour plus d'informations sur la configuration de Jenkins avec le AWS CodeBuild plugin et pour télécharger le code source du plugin, consultez <https://github.com/aws-labs/aws-codebuild-jenkins-plugin>.

Installation du plug-in

Si Jenkins est déjà configuré et que vous souhaitez uniquement installer le plug-in AWS CodeBuild, accédez à votre instance Jenkins, puis dans le gestionnaire de plug-in, recherchez **CodeBuild Plugin for Jenkins**.

Utilisation du plug-in

Pour utiliser AWS CodeBuild avec des sources depuis l'extérieur d'un VPC

1. Créez un projet dans la CodeBuild console. Pour plus d'informations, veuillez consulter [Création d'un projet de génération \(console\)](#).
 - Spécifiez laAWS région sur laquelle exécuter la génération.
 - (Facultatif) Définissez la configuration Amazon VPC pour permettre au conteneur de CodeBuild génération d'accéder aux ressources de votre VPC.
 - Notez le nom de votre projet. Vous en aurez besoin à l'étape 3.
 - (Facultatif) Si votre référentiel source n'est pas pris en charge de manière native par CodeBuild, vous pouvez définir Amazon S3 comme type de source d'entrée pour votre projet.
2. Dans la console IAMConsole, créez un utilisateur à utiliser par le plugin Jenkins.
 - Lorsque vous créez des informations d'identification pour l'utilisateur, choisissez Programmatic Access (Accès par programmation).
 - Créez une stratégie similaire à ce qui suit, puis attachez la stratégie à votre utilisateur.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": ["arn:aws:logs:{{region}}:{{awsAccountId}}: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:{{region}}:{{awsAccountId}}:project/
{{projectName}}"],
    "Action": ["codebuild:StartBuild",
    "codebuild:BatchGetBuilds",
    "codebuild:BatchGetProjects"]
  }
]
}
```

3. Créez un projet freestyle dans Jenkins.

- Sur la page Configure (Configurer), choisissez Add build step (Ajouter une étape de génération), puis choisissez Run build on CodeBuild (Exécuter la génération sur &ACB;).
- Configurez votre étape de génération.
 - Fournissez des valeurs pour Region (Région), Credentials (Informations d'identification) et Project Name (Nom du projet).
 - Choisissez Use Project source (Utiliser la source de projet).
 - Enregistrez la configuration et exécutez une génération à partir de Jenkins.

4. Pour Source Code Management (Gestion de code source), choisissez la façon dont vous souhaitez récupérer votre source. Il se peut que vous deviez installer le GitHub plugin (ou le plugin Jenkins pour votre fournisseur de référentiel source) sur votre serveur Jenkins.

- Sur la page Configure (Configurer), choisissez Add build step (Ajouter une étape de génération), puis choisissez Run build on AWS CodeBuild (Exécuter la génération sur &ACB;).
- Configurez votre étape de génération.
 - Fournissez des valeurs pour Region (Région), Credentials (Informations d'identification) et Project Name (Nom du projet).
 - Choisissez Use Jenkins source (Utiliser la source Jenkins).

- Enregistrez la configuration et exécutez une génération à partir de Jenkins.

Pour utiliser le plug-in AWS CodeBuild avec le plug-in de pipeline Jenkins

- Sur la page de votre projet de pipeline Jenkins, utilisez le générateur d'extraits pour générer un script de pipeline qui s'ajoute en CodeBuild tant qu'étape à votre pipeline. Cela doit générer un script semblable à ce qui suit :

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',  
sourceControlType: 'jenkins'
```

Utilisation d'AWS CodeBuild avec Codecov

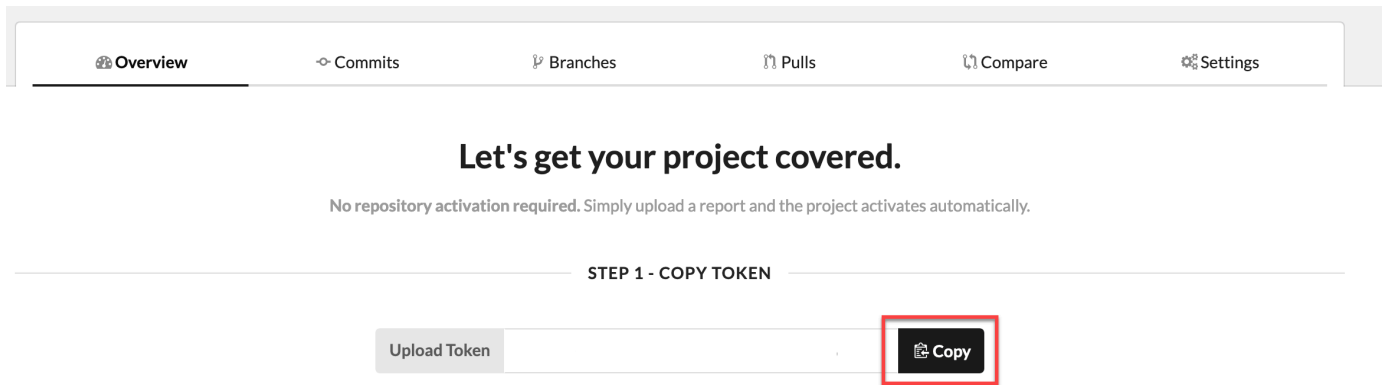
Codecov est un outil qui mesure la couverture de test de votre code. Codecov identifie quelles méthodes et instructions de votre code ne sont pas testées. Utilisez les résultats pour déterminer à quel niveau il pourrait être utile de créer des tests pour améliorer la qualité de votre code. Coecov est disponible pour trois des référentiels source pris en charge par CodeBuild : GitHub, GitHub Enterprise Server et Bitbucket. Si votre projet de génération utilise GitHub Enterprise Server, vous devez utiliser Codecov Enterprise.

Lorsque vous exécutez la génération d'un projet Coecov qui est intégré à Coecov, les rapports Coecov qui analysent le code dans votre référentiel sont chargés dans Coecov. Les journaux de génération comprennent un lien vers ces rapports. Cet exemple montre comment intégrer un projet de génération Python et Java avec Codecov. Pour obtenir la liste des langues prises en charge par Codecov, veuillez consulter [Langues prises en charge par Codecov](#) sur le site web de Codecov.

Intégration de Codecov dans un projet de génération

Pour intégrer Codecov dans votre projet de génération

1. Accédez à <https://codecov.io/signup> et créez un référentiel source GitHub ou Bitbucket. Si vous utilisez GitHub Enterprise, consultez [Codecov Enterprise](#) sur le site web de Codecov.
2. Dans Codecov, ajoutez le référentiel pour lequel vous souhaitez couvrir le code.
3. Lorsque les informations de jeton sont affichées, choisissez Copy (Copier).



- Ajoutez le jeton copié en tant que variable d'environnement `CODECOV_TOKEN` à votre projet de génération. Pour plus d'informations, consultez [Modification des paramètres d'un projet de génération \(console\)](#).
- Créez un fichier texte nommé `my_script.sh` dans votre référentiel. Copiez ce qui suit dans ce fichier :

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

- Choisissez l'onglet Python ou Java selon les utilisations de votre projet de génération, puis procédez comme suit.

Java

- Ajoutez le plug-in Jacoco suivant à `pom.xml` dans votre référentiel.

```
<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>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

```
        <goals>
          <goal>report</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
</plugins>
</build>
```

- Entrez les commandes suivantes dans le fichier buildspec. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).

```
build:
  - mvn test -f pom.xml -fn
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

Python

Entrez les commandes suivantes dans le fichier buildspec. Pour plus d'informations, consultez [Syntaxe d'un fichier buildspec](#).

```
build:
  - pip install coverage
  - coverage run -m unittest discover
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

- Exécutez une version du projet de génération. Un lien vers les rapports Codecov générés pour votre projet apparaît dans les journaux de génération. Utilisez ce lien pour afficher les rapports Codecov. Pour plus d'informations, consultez [Exécution d'une génération dans AWS CodeBuild](#) et [Journalisation des appels d'API AWS CodeBuild avec AWS CloudTrail](#). Les informations Codecov dans les journaux de génération ressemblent à ce qui suit :

```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh
```

```
_____  
/ ____| | | |  
| | | _____| | | | _____ | |
```

```

| | / _ \ / _ ` | / _ \ / _ \ \ / /
| | _ | ( ) | ( | _ / ( | ( ) \ \ /
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \

```

```
Bash-20200303-bc4d7e6
```

```
·[0;90m==>·[0m AWS CodeBuild detected.
```

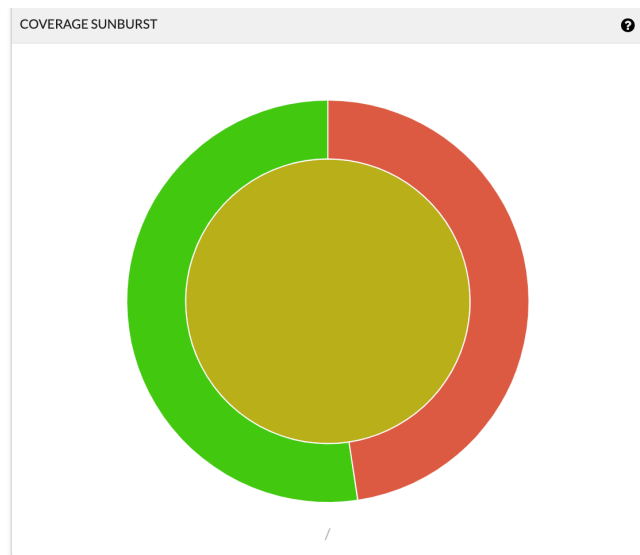
```
... The full list of Codecov log entries has been omitted for brevity ...
```

```
·
```

```
·[0;32m->·[0m View reports at ·[0;36mhttps://codecov.io/github/user/test\_py/commit/commit-id·[0m
```

```
[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

Les rapports se présentent comme suit :



| Files | ≡ | ● | ● | ● | Coverage |
|---------------------------------|----|----|---|---|----------|
| code.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% |

Utiliser AWS CodeBuild avec applications sans serveur

Le AWS Serverless Application Model (AWS SAM) est une infrastructure open source permettant de créer des applications sans serveur. Pour de plus amples informations, veuillez consulter le référentiel GitHub [Modèle d'application serverless AWS](#).

Vous pouvez utiliser AWS CodeBuild pour packuser et déployer des applications sans serveur qui suivent le modèle AWS SAM standard. Pour l'étape de déploiement, CodeBuild peut utiliser AWS CloudFormation. Pour automatiser la génération et le déploiement d'applications sans serveur avec CodeBuild et AWS CloudFormation, vous pouvez utiliser AWS CodePipeline.

Pour de plus amples informations, veuillez consulter [Déploiement d'applications sans serveur](#) dans le AWS Serverless Application Model Manuel du développeur.

Ressources connexes

- Pour plus d'informations sur la mise en route d'AWS CodeBuild, consultez [Mise en route d'AWS CodeBuild à l'aide de la console](#).
- Pour en savoir plus sur le dépannage des problèmes dans CodeBuild, consultez [Résolution des problèmes AWS CodeBuild](#).
- Pour obtenir des informations sur les quotas dans CodeBuild, consultez [Quotas pour AWS CodeBuild](#).

Résolution des problèmes AWS CodeBuild

Utilisez les informations de cette rubrique pour identifier, diagnostiquer et résoudre des problèmes. Pour savoir comment enregistrer et surveiller les CodeBuild versions afin de résoudre les problèmes, consultez [Journalisation et surveillance](#).

Rubriques

- [Apache Maven génère des artefacts de référence à partir du mauvais répertoire](#)
- [Les commandes de génération s'exécutent en tant que racine par défaut](#)
- [Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains Caractères anglais](#)
- [Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store](#)
- [Impossible d'accéder au filtre de branche dans la console CodeBuild](#)
- [Impossible d'afficher la réussite ou l'échec de la génération](#)
- [L'état du build n'a pas été communiqué au fournisseur source](#)
- [Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019](#)
- [Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures](#)
- [Erreur : accès refusé lors de la tentative de téléchargement du cache](#)
- [Erreur : « BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE » lors de l'utilisation d'une image de génération personnalisée](#)
- [Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces](#)
- [Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération](#)
- [Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction](#)
- [Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl »](#)
- [Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération](#)

- [Erreur : « Git clone failed: Unable to access 'your-repository-URL': SSL certificate problem: Self signed certificate »](#)
- [Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération](#)
- [Erreur : « This build image requires selecting at least one runtime version. »](#)
- [Erreur : « QUEUED: INSUFFICIENT_SUBNET » lorsqu'une génération dans une file d'attente échoue](#)
- [Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie »](#)
- [Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied»](#)
- [Erreur : « Unable to locate credentials »](#)
- [RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy](#)
- [Le shell bourne \(sh\) doit exister dans les images de génération](#)
- [Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération](#)
- [Erreur : « Impossible de vérifier l' JobWorker identité » lors de l'ouverture de la CodeBuild console](#)
- [La construction n'a pas pu démarrer](#)
- [Accès aux GitHub métadonnées dans les versions mises en cache localement](#)
- [AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3...](#)

Apache Maven génère des artefacts de référence à partir du mauvais répertoire

Problème : [Lorsque vous utilisez Maven avec un environnement de génération Java AWS CodeBuild fourni, Maven extrait les dépendances de build et de plugin du référentiel central sécurisé de Maven à l'adresse <https://repo1.maven.org/maven2>](#). Cela se produit même si le fichier pom.xml de votre projet de génération déclare explicitement d'autres emplacements à utiliser.

Cause possible : les environnements de génération Java CodeBuild fournis incluent un fichier nommé settings.xml qui est préinstallé dans le répertoire de l'environnement de /root/.m2 construction. Ce fichier settings.xml contient les déclarations suivantes qui demandent à Maven

de toujours extraire les dépendances de génération et de plug-in du référentiel Maven central sécurisé situé à l'adresse <https://repo1.maven.org/maven2>.

```
<settings>
  <activeProfiles>
    <activeProfile>securecentral</activeProfile>
  </activeProfiles>
  <profiles>
    <profile>
      <id>securecentral</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

Solution recommandée : effectuez les opérations suivantes :

1. Ajoutez un fichier `settings.xml` à votre code source.
2. Dans ce fichier `settings.xml`, utilisez le format de fichier `settings.xml` précédent comme guide pour déclarer les référentiels desquels vous voulez que Maven extraie plutôt les dépendances de génération et de plug-in.
3. Au cours de la `install` phase de votre projet de construction, demandez CodeBuild de copier votre `settings.xml` fichier dans le `/root/.m2` répertoire de l'environnement de construction.

Par exemple, examinez l'extrait de code suivant d'un fichier `buildspec.yml` qui illustre ce comportement.

```
version 0.2

phases:
  install:
    commands:
      - cp ./settings.xml /root/.m2/settings.xml
```

Les commandes de génération s'exécutent en tant que racine par défaut

Problème : AWS CodeBuild exécute vos commandes de compilation en tant qu'utilisateur root. Cela se produit même si votre Dockerfile d'image de génération associé définit l'instruction USER sur un autre utilisateur.

Cause : Par défaut, CodeBuild exécute toutes les commandes de compilation en tant qu'utilisateur root.

Solution recommandée : aucune.

Les builds peuvent échouer lorsque les noms de fichiers ne sont pas américains Caractères anglais

Problème : lorsque vous exécutez une version qui utilise des fichiers dont le nom contient des informations non américaines Caractères anglais (par exemple, caractères chinois), la compilation échoue.

Cause possible : les environnements de construction fournis par AWS CodeBuild ont leurs paramètres régionaux par défaut définis sur POSIX. POSIX les paramètres de localisation sont moins compatibles avec les noms de fichiers non américains CodeBuild et les noms de fichiers contenant des informations non américaines Caractères anglais et peuvent entraîner l'échec des versions associées.

Solution recommandée : ajoutez les commandes suivantes à la section `pre_build` de votre fichier `buildspec`. Ces commandes font en sorte que l'environnement de construction utilise l'UTF-8 en

anglais américain pour ses paramètres de localisation, ce qui est plus compatible avec CodeBuild les noms de fichiers contenant des informations non américaines. Caractères anglais.

Pour les environnements de génération basés sur Ubuntu :

```
pre_build:
  commands:
    - export LC_ALL="en_US.UTF-8"
    - locale-gen en_US en_US.UTF-8
    - dpkg-reconfigure locales
```

Pour les environnements de création basés sur Amazon Linux :

```
pre_build:
  commands:
    - export LC_ALL="en_US.utf8"
```

Les builds peuvent échouer lors de l'obtention de paramètres depuis Amazon EC2 Parameter Store

Problème : lorsqu'un build essaie d'obtenir la valeur d'un ou de plusieurs paramètres stockés dans le magasin de paramètres Amazon EC2, le build échoue dans la `DOWNLOAD_SOURCE` phase d'erreur.

`Parameter does not exist`

Cause possible : le rôle de service sur lequel repose le projet de génération n'est pas autorisé à appeler `ssm:GetParameters` ou le projet de génération utilise un rôle de service généré par AWS CodeBuild et permettant d'appeler `ssm:GetParameters`, mais les paramètres ont des noms qui ne commencent pas par `/CodeBuild/`.

Solutions recommandées :

- Si le rôle de service n'a pas été généré par CodeBuild, mettez à jour sa définition CodeBuild pour permettre d'appeler `ssm:GetParameters`. Par exemple, la déclaration de stratégie suivante permet d'appeler l'action `ssm:GetParameters` pour obtenir des paramètres avec des noms commençant par `/CodeBuild/` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Action": "ssm:GetParameters",
    "Effect": "Allow",
    "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/CodeBuild/*"
  }
]
```

- Si le rôle de service a été généré par CodeBuild, mettez à jour sa définition pour autoriser l'accès CodeBuild aux paramètres dans Amazon EC2 Parameter Store avec des noms autres que ceux commençant par `/CodeBuild/`. Par exemple, la déclaration de stratégie suivante permet d'appeler l'action `ssm:GetParameters` pour obtenir des paramètres avec le nom spécifié :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
      "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/PARAMETER_NAME"
    }
  ]
}
```

Impossible d'accéder au filtre de branche dans la console CodeBuild

Problème : L'option de filtre de branche n'est pas disponible dans la console lorsque vous créez ou mettez à jour un AWS CodeBuild projet.

Cause possible : l'option de filtre de branche est obsolète. Elle a été remplacée par les groupes de filtres webhook, qui fournissent davantage de contrôle sur les événements webhook qui déclenchent une nouvelle génération CodeBuild.

Solution recommandée : pour migrer un filtre de branche créé avant l'introduction des filtres webhook, créez un groupes de filtres webhook avec un filtre HEAD_REF avec l'expression régulière `^refs/heads/branchName$`. Par exemple, si votre expression régulière de filtre de branche était `^branchName$`, l'expression régulière mise à jour que vous placez dans le filtre HEAD_REF est `^refs/heads/branchName$`. Pour plus d'informations, consultez [Événements du webhook Bitbucket](#) et [Filtrer les événements du GitHub webhook \(console\)](#).

Impossible d'afficher la réussite ou l'échec de la génération

Problème : vous ne pouvez pas consulter l'état de réussite ou d'échec lorsqu'une génération a fait l'objet d'une nouvelle tentative.

Cause possible : l'option qui permet de signaler le statut de votre génération n'est pas activée.

Solutions recommandées : Activez le statut de génération du rapport lorsque vous créez ou mettez à jour un CodeBuild projet. Cette option indique à CodeBuild de renvoyer l'état lorsque vous déclenchez une génération. Pour plus d'informations, consultez [reportBuildStatus](#) dans la Référence d'API AWS CodeBuild .

L'état du build n'a pas été communiqué au fournisseur source

Problème : Après avoir autorisé le signalement de l'état du build à un fournisseur source, tel que GitHub Bitbucket, le statut du build n'est pas mis à jour.

Cause possible : L'utilisateur associé au fournisseur de source n'a pas accès en écriture au dépôt.

Solutions recommandées : Pour pouvoir signaler l'état de construction au fournisseur de source, l'utilisateur associé au fournisseur de source doit avoir un accès en écriture au dépôt. Si l'utilisateur ne dispose pas d'un accès en écriture, l'état de construction ne peut pas être mis à jour. Pour de plus amples informations, veuillez consulter [Accès au fournisseur de source](#).

Impossible de trouver et de sélectionner l'image de base de la plate-forme Windows Server Core 2019

Problème : Impossible de trouver ou de sélectionner l'image de base de la plateforme Windows Server Core 2019.

Cause possible : vous utilisez une AWS région qui ne prend pas en charge cette image.

Solutions recommandées : utilisez l'une des AWS régions suivantes où l'image de base de la plateforme Windows Server Core 2019 est prise en charge :

- USA Est (Virginie du Nord)
- USA Est (Ohio)

- USA Ouest (Oregon)
- Europe (Irlande)

Les commandes antérieures des fichiers buildspec ne sont pas reconnues par les commandes ultérieures

Problème : les résultats d'une ou plusieurs commandes dans votre fichier buildspec ne sont pas reconnus par des commandes ultérieures dans le même fichier buildspec. Par exemple, une commande peut définir une variable d'environnement local, mais une commande exécutée ultérieurement ne parvient pas à obtenir la valeur de cette variable d'environnement local.

Cause possible : dans le fichier buildspec version 0.1, AWS CodeBuild exécute chaque commande dans une instance distincte du shell par défaut dans l'environnement de génération. Cela signifie que chaque commande s'exécute isolée de toutes les autres commandes. Par défaut, vous ne pouvez pas exécuter une commande unique qui s'appuie sur l'état de commandes précédentes.

Solutions recommandées : nous vous recommandons d'utiliser la version 0.2 de buildspec afin de résoudre ce problème. Si vous devez utiliser la version 0.1 de buildspec pour une raison quelconque, nous vous recommandons d'exploiter l'opérateur de chaînage de commande shell (par exemple, `&&` dans Linux) pour combiner plusieurs commandes en une seule. Ou incluez dans votre code source un script shell contenant plusieurs commandes, puis appelez ce script shell à partir d'une commande unique dans le fichier buildspec. Pour plus d'informations, consultez [Shells et commandes dans les environnements de génération](#) et [Variables d'environnement dans les environnements de génération](#).

Erreur : accès refusé lors de la tentative de téléchargement du cache

Problème : lors de la tentative de téléchargement du cache sur un projet de génération dont le cache est activé, vous recevez l'erreur `Access denied`.

Causes possibles :

- vous venez de configurer la mise en cache dans le cadre de votre projet de génération.
- Le cache a récemment été invalidé par le biais de l'API `InvalidateProjectCache`.
- Le rôle de service utilisé par CodeBuild n'a aucune `s3:PutObject` autorisation `s3:GetObject` d'accès au compartiment S3 qui contient le cache.

Solution recommandée : pour la première utilisation, il est normal de voir cela immédiatement après la mise à jour de la configuration du cache. Si cette erreur persiste, vérifiez si le rôle de service a les autorisations `s3:GetObject` et `s3:PutObject` pour le compartiment S3 qui contient le cache. Pour plus d'informations, consultez [Spécifier les autorisations S3](#) dans le manuel Amazon S3 Developer Guide.

Erreur : « BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE » lors de l'utilisation d'une image de génération personnalisée

Problème : lorsque vous essayez d'exécuter une génération qui utilise une version de génération personnalisée, la génération échoue avec l'erreur BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE.

Cause possible : la taille globale non compressée de l'image de génération est supérieure à l'espace disque disponible du type de calcul de l'environnement de génération. Pour vérifier la taille de votre image de génération, utilisez Docker pour exécuter la commande `docker images REPOSITORY:TAG`. Pour obtenir une liste de l'espace disque disponible par type de calcul, reportez-vous à la section [Modes et types de calcul de l'environnement de création](#).

Solution recommandée : utilisez un type de calcul plus important avec plus d'espace disque disponible, ou réduisez la taille de votre image de construction personnalisée.

Cause possible : AWS CodeBuild n'est pas autorisé à extraire l'image de compilation de votre Amazon Elastic Container Registry (Amazon ECR).

Solution recommandée : mettez à jour les autorisations de votre référentiel dans Amazon ECR afin de CodeBuild pouvoir intégrer votre image de construction personnalisée dans l'environnement de création. Pour plus d'informations, consultez le [Échantillon Amazon ECR](#).

Cause possible : L'image Amazon ECR que vous avez demandée n'est pas disponible dans la AWS région utilisée par votre AWS compte.

Solution recommandée : utilisez une image Amazon ECR située dans la même AWS région que celle utilisée par votre AWS compte.

Cause possible : vous utilisez un registre privé dans un VPC qui ne dispose pas d'un accès public à Internet. CodeBuild Impossible d'extraire une image d'une adresse IP privée dans un VPC. Pour de plus amples informations, veuillez consulter [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#).

Solution recommandée : Si vous utilisez un registre privé dans un VPC, assurez-vous que celui-ci dispose d'un accès public à Internet.

Cause possible : si le message d'erreur contient `toomanyrequests` et que l'image provient de Docker Hub, cette erreur signifie que la limite d'attraction de Docker Hub a été atteinte.

Solution recommandée : utilisez un registre privé Docker Hub ou procurez-vous votre image auprès d'Amazon ECR. Pour plus d'informations sur l'utilisation d'un registre privé, consultez [Registre privé avec AWS Secrets Manager échantillon pour CodeBuild](#). Pour plus d'informations sur l'utilisation d'Amazon ECR, consultez [Exemple Amazon ECR pour CodeBuild](#).

Erreur : « Le conteneur de construction a été trouvé mort avant de terminer la construction. Le conteneur de construction est mort parce qu'il n'y avait plus de mémoire ou parce que l'image Docker n'est pas prise en charge. ErrorCode: 500 pouces

Problème : Lorsque vous essayez d'utiliser un conteneur Microsoft Windows ou Linux dans AWS CodeBuild, cette erreur se produit pendant la phase de PROVISIONING.

Causes possibles :

- La version du système d'exploitation du conteneur n'est pas prise en charge par CodeBuild.
- `HTTP_PROXY`, `HTTPS_PROXY`, ou les deux sont spécifiés dans le conteneur.

Solutions recommandées :

- Pour Microsoft Windows, utilisez un conteneur Windows avec une version de système d'exploitation de conteneur `microsoft/windowsservercore:10.0.x` (par exemple, `microsoft/windowsservercore:10.0.14393.2125`).
- Pour Linux, effacez les paramètres `HTTP_PROXY` et `HTTPS_PROXY` de votre image Docker, ou spécifiez la configuration du VPC dans votre projet.

Erreur : « Cannot connect to the Docker daemon » lors de l'exécution d'une génération

Problème : la génération échoue et vous recevez une erreur similaire à `Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?` dans le journal de génération.

Cause possible : vous n'exécutez pas la génération en mode privilégié.

Solution recommandée : pour corriger cette erreur, vous devez activer le mode privilégié et mettre à jour votre buildspec en suivant les instructions suivantes.

Pour exécuter votre build en mode privilégié, procédez comme suit :

1. Ouvrez la CodeBuild console à l'[adresse https://console.aws.amazon.com/codebuild/](https://console.aws.amazon.com/codebuild/).
2. Dans le volet de navigation, choisissez Build projects, puis choisissez votre projet de build.
3. Dans Modifier, choisissez Environnement.
4. Sélectionnez Additional configuration (Configuration supplémentaire).
5. Dans Privileged, sélectionnez Activer cet indicateur si vous souhaitez créer des images Docker ou si vous souhaitez que vos builds bénéficient de privilèges élevés. .
6. Choisissez Mettre à jour l'environnement.
7. Choisissez Démarrer la génération pour réessayer de créer votre génération.

Vous devrez également démarrer le démon Docker à l'intérieur de votre conteneur. La `install` phase de votre buildspec peut ressembler à ceci.

```
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Pour plus d'informations sur le pilote de stockage OverlayFS référencé dans le fichier buildspec, consultez [Utilisation du pilote de stockage OverlayFS](#) sur le site web Docker.

Note

Si le système d'exploitation de base est Alpine Linux, dans `buildspec.yml` ajoutez l'argument `-t` à `timeout` :

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Pour en savoir plus sur la création et l'exécution d'une image Docker à l'aide AWS CodeBuild de [Docker dans un exemple d'image personnalisé pour CodeBuild](#)

Erreur : « n'CodeBuild est pas autorisé à exécuter : sts : AssumeRole » lors de la création ou de la mise à jour d'un projet de construction

Problème : lorsque vous essayez de créer ou de mettre à jour un projet de génération, vous recevez l'erreur `Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name`.

Causes possibles :

- Le AWS Security Token Service (AWS STS) a été désactivé pour la AWS région dans laquelle vous essayez de créer ou de mettre à jour le projet de construction.
- Le rôle AWS CodeBuild de service associé au projet de construction n'existe pas ou ne dispose pas d'autorisations suffisantes pour être approuvé CodeBuild.

Solutions recommandées :

- Assurez-vous qu' AWS STS il est activé pour la AWS région dans laquelle vous essayez de créer ou de mettre à jour le projet de construction. Pour plus d'informations, consultez la section [Activation et désactivation AWS STS dans une AWS région](#) dans le guide de l'utilisateur IAM.
- Assurez-vous que le rôle CodeBuild de service cible existe dans votre AWS compte. Si vous n'utilisez pas la console, vérifiez que vous avez bien orthographié l'Amazon Resource Name (ARN) du rôle de service lorsque vous avez créé ou mis à jour le projet de génération.
- Assurez-vous que le rôle de CodeBuild service cible dispose d'autorisations suffisantes pour être fiable CodeBuild. Pour plus d'informations, consultez la déclaration de stratégie de relation d'approbation dans [Création d'un rôle CodeBuild de service](#).

Erreur : « Erreur lors de l'appel GetBucketAcl : soit le propriétaire du compartiment a changé, soit le rôle de service n'est plus autorisé à appeler s3 : GetBucketAcl »

Problème : lorsque vous exécutez une génération, vous recevez une erreur sur une modification de la propriété d'un compartiment S3 et d'autorisations GetBucketAcl.

Cause possible : vous avez ajouté les `s3:GetBucketLocation` autorisations `s3:GetBucketAcl` et à votre rôle IAM. Ces autorisations protègent le compartiment S3 de votre projet et garantissent que vous seul pouvez y accéder. Après l'ajout de ces autorisations, le propriétaire du compartiment S3 a changé.

Solution recommandée : vérifiez que vous êtes propriétaire du compartiment S3, puis ajoutez de nouveau des autorisations à votre rôle IAM. Pour de plus amples informations, veuillez consulter [Accès sécurisé aux compartiments S3](#).

Erreur : « Failed to upload artifacts: Invalid arn » lors de l'exécution d'une génération

Problème : lorsque vous exécutez une génération, la phase `UPLOAD_ARTIFACTS` de la génération échoue avec l'erreur `Failed to upload artifacts: Invalid arn`.

Cause possible : votre compartiment de sortie S3 (le compartiment AWS CodeBuild dans lequel sont stockés les résultats de la génération) se trouve dans une AWS région différente de celle du projet de CodeBuild construction.

Solution recommandée : mettez à jour les paramètres du projet de construction pour qu'ils pointent vers un compartiment de sortie situé dans la même AWS région que le projet de construction.


Erreur : « Git clone failed: Unable to access 'your-repository-URL**': SSL certificate problem: Self signed certificate »**

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Cause possible : Votre référentiel source possède un certificat auto-signé, mais vous n'avez pas choisi d'installer le certificat depuis votre compartiment S3 dans le cadre de votre projet de génération.

Solutions recommandées :

- modifiez votre projet. Pour Certificat, choisissez Installer le certificat à partir de votre compartiment S3. Pour Compartiment de certificat, choisissez le compartiment S3 dans lequel votre certificat SSL est stocké. Pour Object key of certificate (Clé d'objet de certificat), tapez le nom de la clé d'objet S3.
- modifiez votre projet. Sélectionnez SSL non sécurisé pour ignorer les avertissements SSL lors de la connexion au référentiel de votre projet GitHub Enterprise Server.

 Note

Nous vous recommandons de n'utiliser Insecure SSL que pour les tests. Cette option ne doit pas être utilisée dans un environnement de production.

Erreur : « The bucket you are attempting to access must be addressed using the specified endpoint... » lors de l'exécution d'une génération

Problème : lorsque vous exécutez une génération, la phase `DOWNLOAD_SOURCE` de la génération échoue avec l'erreur `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

Cause possible : votre code source prédéfini est stocké dans un compartiment S3, et ce compartiment se trouve dans une AWS région différente de celle du projet de AWS CodeBuild génération.

Solution recommandée : mettez à jour les paramètres du projet de génération pour pointer vers un compartiment qui contient votre code source précompilé. Assurez-vous que le bucket se trouve dans la même AWS région que le projet de construction.

Erreur : « This build image requires selecting at least one runtime version. »

Problème : lorsque vous exécutez une génération, la phase `DOWNLOAD_SOURCE` de la génération échoue avec l'erreur `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

Cause possible : votre génération utilise la version 1.0 ou ultérieure de l'image standard Amazon Linux 2 (AL2), ou la version 2.0 ou ultérieure de l'image standard Ubuntu, et aucun environnement d'exécution n'est spécifié dans le fichier `buildspec`.

Solution recommandée : Si vous utilisez l'image `aws/codebuild/standard:2.0` CodeBuild gérée, vous devez spécifier une version d'exécution dans la `runtime-versions` section du fichier `buildspec`. Par exemple, vous pouvez utiliser le fichier `buildspec` suivant pour un projet qui utilise PHP :

```
version: 0.2

phases:
  install:
    runtime-versions:
      php: 7.3
  build:
    commands:
      - php --version
artifacts:
  files:
    - README.md
```

Note

Si vous spécifiez une `runtime-versions` section et utilisez une image autre qu'Ubuntu Standard Image 2.0 ou version ultérieure, ou l'image standard Amazon Linux 2 (AL2) 1.0 ou version ultérieure, le build émet l'avertissement « »`Skipping install of runtimes. Runtime version selection is not supported by this build image.`

Pour de plus amples informations, veuillez consulter [Specify runtime versions in the buildspec file.](#)

Erreur : « QUEUED: INSUFFICIENT_SUBNET » lorsqu'une génération dans une file d'attente échoue

Problème : une génération dans une file d'attente échoue avec une erreur similaire à QUEUED: INSUFFICIENT_SUBNET.

Causes possibles : le bloc d'adresse CIDR IPv4 spécifié pour votre VPC utilise une adresse IP réservée. Les quatre premières adresses IP et la dernière adresse IP du bloc d'adresse CIDR de chaque sous-réseau ne sont pas disponibles pour utilisation, et ne peuvent donc pas être affectées à une instance. Par exemple, dans un sous-réseau avec le bloc d'adresse CIDR 10.0.0.0/24, les cinq adresses IP suivantes sont réservées :

- 10.0.0.0 : Adresse réseau.
- 10.0.0.1: Réserve par AWS pour le routeur VPC.
- 10.0.0.2: Réserve par AWS. L'adresse IP du serveur DNS est toujours la base de la plage d'adresses du réseau VPC, plus deux ; cependant, nous réservons également la base de chaque sous-réseau, plus deux. Pour les VPC ayant plusieurs blocs CIDR, l'adresse IP du serveur DNS se trouve dans le CIDR principal. Pour plus d'informations, consultez [Serveur Amazon DNS](#) dans le Amazon VPC Guide de l'utilisateur.
- 10.0.0.3: Réserve par AWS pour une utilisation future.
- 10.0.0.255 : Adresse de diffusion réseau. Nous ne prenons pas en charge la diffusion dans un VPC. Cette adresse est réservée.

Solutions recommandées : vérifiez si votre VPC utilise une adresse IP réservée. Remplacez toute adresse IP réservée par une adresse non réservée. Pour plus d'informations, consultez [Dimensionnement des VPC et des sous-réseaux](#) dans le Amazon VPC Guide de l'utilisateur.

Erreur : « Impossible de télécharger le cache RequestError : échec de l'envoi de la demande en raison de : x509 : échec du chargement des racines du système et aucune racine n'a été fournie »

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Cause possible : vous avez configuré la mise en cache dans le cadre de votre projet de génération et vous utilisez une ancienne image Docker qui inclut un certificat racine expiré.

Solution recommandée : mettez à jour l'image Docker utilisée dans votre AWS CodeBuild projet. Pour de plus amples informations, veuillez consulter [Images Docker fournies par CodeBuild](#).

Erreur : « Impossible de télécharger le certificat depuis S3. AccessDenied »

Problème : lorsque vous essayez d'exécuter un projet de génération, la génération échoue avec cette erreur.

Causes possibles :

- vous avez choisi le mauvais compartiment S3 pour votre certificat.
- Vous avez entré la mauvaise clé d'objet pour votre certificat.

Solutions recommandées :

- modifiez votre projet. Pour Compartiment de certificat, choisissez le compartiment S3 dans lequel votre certificat SSL est stocké.
- modifiez votre projet. Pour Object key of certificate (Clé d'objet de certificat), tapez le nom de la clé d'objet S3.

Erreur : « Unable to locate credentials »

Problème : Lorsque vous essayez d'exécuter le AWS CLI, d'utiliser un AWS SDK ou d'appeler un autre composant similaire dans le cadre d'une compilation, vous obtenez des erreurs de génération directement liées au AWS CLI AWS SDK ou au composant. Par exemple, vous pouvez obtenir l'erreur de génération `Unable to locate credentials`.

Causes possibles :

- La version du AWS CLI AWS SDK ou du composant de l'environnement de construction est incompatible avec AWS CodeBuild.
- Vous exécutez un conteneur Docker dans un environnement de génération qui utilise Docker, et le conteneur n'a pas accès aux AWS informations d'identification par défaut.

Solutions recommandées :

- Assurez-vous que votre environnement de génération dispose de la version suivante ou supérieure du AWS CLI AWS SDK ou du composant.
 - AWS CLI : 1.10.47
 - AWS SDK pour C++ : 0.2.19
 - AWS SDK pour Go : 1.2.5
 - AWS SDK pour Java : 1.11.16
 - AWS SDK pour JavaScript : 2.4.7
 - AWS SDK pour PHP : 3.18.28
 - AWS SDK pour Python (Boto3) : 1.4.0
 - AWS SDK pour Ruby : 2.3.22
 - Botocore : 1.4.37
 - CoreCLR : 3.2.6-beta
 - Node.js : 2.4.7
- Si vous devez exécuter un conteneur Docker dans un environnement de génération et que le conteneur nécessite des AWS informations d'identification, vous devez transmettre les informations d'identification de l'environnement de génération au conteneur. Dans le fichier buildspec, incluez une commande Docker `run` telle que la suivante. Cet exemple utilise la commande `aws s3 ls` pour répertorier les compartiments S3 disponibles. L'`-e` option passe par les variables d'environnement requises pour que votre conteneur puisse accéder aux AWS informations d'identification.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Si vous créez une image Docker et que la compilation nécessite des AWS informations d'identification (par exemple, pour télécharger un fichier depuis Amazon S3), vous devez transmettre les informations d'identification de l'environnement de génération au processus de génération Docker comme suit.
 1. Dans votre Dockerfile de code source pour l'image Docker, spécifiez les instructions ARG suivantes.

```
ARG AWS_DEFAULT_REGION
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. Dans le fichier `buildspec`, incluez une commande `Docker build` telle que la suivante. Les `--build-arg` options définissent les variables d'environnement requises pour que votre processus de création de Docker puisse accéder aux AWS informations d'identification.

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg  
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -  
t your-image-tag .
```

RequestError erreur de temporisation lors de l'exécution CodeBuild sur un serveur proxy

Problème : Vous recevez une erreur `RequestError` similaire à l'une des erreurs suivantes :

- `RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeout` CloudWatch Logs.
- `Error uploading artifacts: RequestError: send request failed caused by: Put https://your-bucket.s3.your-aws-region.amazonaws.com/*: dial tcp 52.219.96.208:443: connect: connection refused` depuis Amazon S3.

Causes possibles :

- `ssl-bump` n'est pas configuré correctement.
- La stratégie de sécurité de votre organisation ne vous permet pas d'utiliser `ssl_bump`.
- Votre fichier `buildspec` n'a pas de paramètres proxy spécifiés à l'aide d'un élément `proxy`.

Solutions recommandées :

- assurez-vous que `ssl-bump` est configuré correctement. Si vous utilisez Squid pour votre serveur proxy, consultez [Configuration de Squid en tant que serveur proxy explicite](#).
- Pour utiliser des points de terminaison privés pour Amazon S3 et CloudWatch Logs, procédez comme suit :
 1. Dans la table de routage de votre sous-réseau privé, supprimez la règle que vous avez ajoutée qui achemine le trafic destiné à Internet vers votre serveur proxy. Pour plus d'informations,

- consultez [la section Création d'un sous-réseau dans votre VPC](#) dans le guide de l'utilisateur Amazon VPC.
2. Créez un point de terminaison Amazon S3 et un point de terminaison CloudWatch Logs privés et associez-les au sous-réseau privé de votre Amazon VPC. Pour plus d'informations, consultez la section [Services de point de terminaison VPC](#) dans le guide de l'utilisateur Amazon VPC.
 3. Vérifiez que l'option Activer le nom DNS privé dans votre Amazon VPC est sélectionnée. Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.
- Si vous n'utilisez pas `ssl-bump` pour un serveur proxy explicite, ajoutez une configuration proxy à votre fichier `buildspec` à l'aide d'un élément `proxy`. Pour plus d'informations, consultez [Exécution de CodeBuild sur un serveur proxy explicite](#) et [Syntaxe d'un fichier buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
```

Le shell bourne (sh) doit exister dans les images de génération

Problème : vous utilisez une image de build qui n'est pas fournie par AWS CodeBuild, et vos builds échouent avec le messageBuild `container found dead before completing the build`.

Cause possible : Le shell Bourne (sh) n'est pas inclus dans votre image de construction. CodeBuild doit sh exécuter des commandes et des scripts de construction.

Solution recommandée : S'il sh n'est pas présent dans votre image de construction, assurez-vous de l'inclure avant de commencer toute autre génération utilisant votre image. (inclut CodeBuild déjà des images sh dans sa version.)

Avertissement : « Skipping install of runtimes. runtime version selection is not supported by this build image » lors de l'exécution d'une génération

Problème : lorsque vous exécutez une génération, le journal de génération contient cet avertissement.

Cause possible : votre build n'utilise pas la version 1.0 ou ultérieure de l'image standard Amazon Linux 2 (AL2), ni la version 2.0 ou ultérieure de l'image standard Ubuntu, et un environnement d'exécution est spécifié dans une `runtime-versions` section de votre fichier `buildspec`.

Solution recommandée : vérifiez que votre fichier `buildspec` ne contient pas une section `runtime-versions`. La `runtime-versions` section n'est obligatoire que si vous utilisez l'image standard Amazon Linux 2 (AL2) ou version ultérieure ou l'image standard Ubuntu version 2.0 ou ultérieure.

Erreur : « Impossible de vérifier l' JobWorker identité » lors de l'ouverture de la CodeBuild console

Problème : Lorsque vous ouvrez la CodeBuild console, le message d'erreur « Impossible de vérifier l' JobWorker identité » s'affiche.

Cause possible : le rôle IAM utilisé pour accéder à la console possède une balise utilisée `jobId` comme clé. Cette clé de balise est réservée CodeBuild et provoquera cette erreur si elle est présente.

Solution recommandée : modifiez toutes les balises de rôle IAM personnalisées contenant la clé `jobId` pour qu'elles aient une clé différente, par exemple `jobIdentifier`.

La construction n'a pas pu démarrer

Problème : Lorsque vous démarrez une compilation, vous recevez un message d'erreur indiquant que la compilation n'a pas pu démarrer.

Cause possible : le nombre de versions simultanées a été atteint.

Solutions recommandées : attendez que les autres versions soient terminées ou augmentez la limite de génération simultanée pour le projet, puis recommencez la génération. Pour de plus amples informations, veuillez consulter [Configuration du projet](#).

Accès aux GitHub métadonnées dans les versions mises en cache localement

Problème : Dans certains cas, le répertoire `.git` d'une version mise en cache est un fichier texte et non un répertoire.

Causes possibles : lorsque la mise en cache des sources locales est activée pour une compilation, CodeBuild crée un gitlink pour le `.git` répertoire. Cela signifie que le `.git` répertoire est en fait un fichier texte contenant le chemin d'accès au répertoire.

Solutions recommandées : Dans tous les cas, utilisez la commande suivante pour obtenir le répertoire de métadonnées Git. Cette commande fonctionnera quel que soit le format de `.git` :

```
git rev-parse --git-dir
```

AccessDenied: Le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment S3...

Problème : lors du téléchargement de données de test dans un compartiment Amazon S3, CodeBuild il est impossible d'écrire les données de test dans le compartiment.

Causes possibles :

- Le compte spécifié pour le propriétaire du compartiment du groupe de rapports ne correspond pas au propriétaire du compartiment Amazon S3.
- Le rôle de service ne dispose pas d'un accès en écriture au compartiment.

Solutions recommandées :

- Modifiez le propriétaire du compartiment du groupe de rapports pour qu'il corresponde au propriétaire du compartiment Amazon S3.
- Modifiez le rôle de service pour autoriser l'accès en écriture au compartiment Amazon S3.

Quotas pour AWS CodeBuild

Les tableaux suivants répertorient les quotas actuels d'AWS CodeBuild. Ces quotas s'appliquent à chaque région AWS prise en charge pour chaque compte AWS, sauf indication contraire.

Quotas de service

Les quotas par défaut pour le AWS CodeBuild service sont les suivants.

| Nom | Par défaut | Ajusté | Description |
|---|--|---------------------|--|
| Tags associés par projet | Chaque région prise en charge : 50 | Non | Nombre maximum de balises que vous pouvez associer à un projet de génération |
| Projets de génération | Chaque région prise en charge : 5 000 | Oui | Nombre maximal de projets de version |
| Délai d'expiration du délai d'attente en minutes | Chaque région prise en charge : 480 | Non | Délai de construction maximal en minutes |
| Demande simultanée d'informations sur les builds | Chaque Région prise en charge : 100 | Non | Nombre maximum de versions pour lesquelles vous pouvez demander des informations à tout moment à l'aide de la AWS CLI ou d'un AWS SDK. |
| Demandes simultanées d'informations sur les projets de construction | Chaque Région prise en charge : 100 | Non | Nombre maximum de projets de construction pour lesquels vous pouvez demander des informations à tout |

| Nom | Par défaut | Ajuste | Description |
|--|--------------------------------|---------------------|--|
| | | | moment à l'aide de la AWS CLI ou d'un AWS SDK. |
| Exécution simultanée de versions pour l'environnement ARM Lambda/10 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/10 Go |
| Exécution simultanée de versions pour l'environnement ARM Lambda/1 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/1 Go |
| Exécution simultanée de versions pour l'environnement ARM Lambda/2 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/2 Go |
| Exécution simultanée de versions pour l'environnement ARM Lambda/4 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/4 Go |
| Exécution simultanée de versions pour l'environnement ARM Lambda/8 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement ARM Lambda/8 Go |

| Nom | Par défaut | Ajusté | Description |
|--|-----------------------------------|---------------------|--|
| Exécution simultanée de versions pour un environnement ARM/Large | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour un environnement ARM/Large |
| Exécution simultanée de versions pour un environnement ARM/Small | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour un environnement ARM/Small |
| Exécution simultanée de versions pour un environnement Linux GPU Large | Chaque région prise en charge : 0 | Oui | Nombre maximum de versions exécutées simultanément pour un environnement GPU/Large Linux |
| Exécution simultanée de versions pour un environnement Linux GPU Small | Chaque région prise en charge : 0 | Oui | Nombre maximum de versions exécutées simultanément pour le GPU/environnement Small Linux |
| Exécution simultanée de versions pour l'environnement Linux Lambda/10 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/10 Go |
| Exécution simultanée de versions pour l'environnement Linux Lambda/1 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/1 Go |

| Nom | Par défaut | Ajusté | Description |
|---|-----------------------------------|---------------------|---|
| Exécution simultanée de versions pour l'environnement Linux Lambda/2 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/2 Go |
| Exécution simultanée de versions pour l'environnement Linux Lambda/4 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/4 Go |
| Exécution simultanée de versions pour l'environnement Linux Lambda/8 Go | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux Lambda/8 Go |
| Exécution simultanée de versions pour l'environnement Linux/2xLarge | Chaque région prise en charge : 0 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/2xLarge |
| Exécution simultanée de versions pour l'environnement Linux/Large | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Large |
| Exécution simultanée de versions pour l'environnement Linux/Medium | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Medium |

| Nom | Par défaut | Ajusté | Description |
|---|--------------------------------|---------------------|--|
| Exécution simultanée de versions pour l'environnement Linux/Small | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/Small |
| Exécution simultanée de versions pour l'environnement Linux/XLarge | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Linux/XLarge |
| Exécution simultanée de versions pour l'environnement Windows Server 2019/ Large | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2019/Large |
| Exécution simultanée de versions pour l'environnement Windows Server 2019/ Medium | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Windows Server 2019/Medium |
| Exécution simultanée de versions pour un environnement Windows/Large | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour un environnement Windows/ Large |
| Exécution simultanée de versions pour l'environnement Windows/Medium | Par région prise en charge : 1 | Oui | Nombre maximum de versions exécutées simultanément pour l'environnement Windows/Medium |

| Nom | Par défaut | Ajuste | Description |
|---|--|--------|---|
| Période minimale d'expiration de la construction en minutes | Chaque région prise en charge : 5 | Non | Délai de construction minimal en minutes |
| Groupes de sécurité sous configuration VPC | Chaque région prise en charge : 5 | Non | Groupes de sécurité disponibles pour la configuration VPC |
| Sous-réseaux sous configuration VPC | Chaque région prise en charge : 16 | Non | Sous-réseaux disponibles pour la configuration VPC |

Note

Les métriques internes détermineront les quotas par défaut pour les builds exécutés simultanément.

Les quotas pour le nombre maximum de builds exécutés simultanément varient en fonction du type de calcul. Pour certaines plateformes et certains types de calcul, la valeur par défaut est 20. Pour demander un quota de construction simultanée plus élevé, ou si le message d'erreur « Impossible d'avoir plus de X versions actives pour le compte » s'affiche, utilisez le lien ci-dessus pour effectuer la demande. Pour plus d'informations sur les tarifs, consultez la section [AWS CodeBuildtarification](#).

Autres limites

Projets de génération

| Ressource | Par défaut |
|---|---|
| Caractères autorisés dans une description de projet de génération | N'importe quel compte |
| Caractères autorisés dans un nom de projet de génération | Les lettres A-Z et a-z, les chiffres 0-9, et les caractères spéciaux - et _ |

| Ressource | Par défaut |
|---|----------------------------|
| Longueur d'un nom de projet de génération | 2 à 255 caractères, inclus |
| Longueur maximum d'une description de projet de génération | 255 caractères |
| Nombre maximal de rapports que vous pouvez ajouter à un projet | 5 |
| Nombre de minutes que vous pouvez spécifier comme délai d'expiration de génération dans un projet de génération pour toutes les générations associées | 5 à 480 (8 heures) |

Versions

| Ressource | Par défaut |
|---|--------------------|
| Durée maximale de conservation de l'historique d'une génération | 1 an |
| Nombre de minutes que vous pouvez spécifier comme délai d'expiration de génération dans une génération unique | 5 à 480 (8 heures) |

Flottes informatiques

| Ressource | Par défaut |
|--|------------|
| Nombre simultané de flottes informatiques | 10 |
| Instances exécutées simultanément pour les flottes ARM/Small Environment | 1 |

| Ressource | Par défaut |
|---|------------|
| Instances exécutées simultanément pour les flottes d'environnements ARM/Large | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Linux/Small | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Linux/Medium | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Linux/Large | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Linux/XLarge | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Linux/2XLarge | 0 |
| Instances exécutées simultanément pour les flottes de GPU Linux/de petits environnements | 0 |
| Instances exécutées simultanément pour les GPU/flottes d'environnements de grande taille sous Linux | 0 |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2019/Medium | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2019/Large | 1 |
| Instances exécutées simultanément pour les flottes d'environnements Windows Server 2022/Medium | 1 |

| Ressource | Par défaut |
|--|------------|
| Instances exécutées simultanément pour Windows Server 2022/grands parcs d'environnements | 1 |

Rapports

| Ressource | Par défaut |
|--|------------------|
| Durée maximale d'un rapport de test est disponible après sa création | 30 jours |
| Longueur maximale d'un message de scénario de test | 5 000 caractères |
| Longueur maximale du nom d'un scénario de test | 1 000 caractères |
| Nombre maximal de groupes de rapports par compte AWS | 5 000 |
| Nombre maximal de cas de test par rapport au rapport | 500 |

Balises

Les limites de balises s'appliquent aux balises figurant sur les projets de CodeBuild création et les ressources des groupes de CodeBuild rapports.

| Ressource | Par défaut |
|-------------------------------------|--|
| Noms de clés de balise de ressource | Toute combinaison de lettres, chiffres, espaces au format Unicode et caractères UTF-8 autorisés avec une longueur de 1 à |

| Ressource | Par défaut |
|------------------------------------|--|
| | <p>127 caractères. Les caractères autorisés sont les suivants : + - = . _ : / @</p> <p>Les noms de clé de balise doivent être uniques, et chaque clé ne peut avoir qu'une seule valeur. Un nom de clé de balise ne peut pas :</p> <ul style="list-style-type: none"> • commencer par aws : • être composée uniquement d'espaces • se terminer par un espace • contenir des émoticônes ou l'un des caractères suivants : ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ; |
| Valeurs de la balise de ressource. | <p>Toute combinaison de lettres, chiffres, espaces au format Unicode et caractères UTF-8 autorisés avec une longueur de 0 à 255 caractères. Les caractères autorisés sont les suivants : + - = . _ : / @</p> <p>Une clé ne peut avoir qu'une seule valeur, mais plusieurs clés peuvent avoir la même valeur. Une valeur de clé de balise ne peut pas contenir d'émoticônes ou l'un des caractères suivants : ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;</p> |

Notices de tiers AWS CodeBuild pour Windows

Lorsque vous utilisez des versions CodeBuild pour Windows, vous avez la possibilité d'utiliser des packages et modules tiers pour permettre à votre application créée de s'exécuter sur les systèmes d'exploitation Microsoft Windows et d'interagir avec certains produits tiers. La liste suivante contient les conditions d'utilisation légales tierces qui régissent votre utilisation des packages/modules tiers spécifiés.

Rubriques

- [1\) image Docker de base — Windowsservercore](#)
- [2\) Image Docker basée sur Windows : choco](#)
- [3\) Image Docker basée sur Windows : git --version 2.16.2](#)
- [4\) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools](#)
- [5\) Image Docker basée sur Windows : nuget.commandline --version 4.5.1](#)
- [7\) Image Docker basée sur Windows : netfx-4.6.2-devpack](#)
- [8\) Image Docker basée sur Windows — visualfsharptools, version 4.0](#)
- [9\) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies](#)
- [10\) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1](#)
- [11\) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx](#)
- [12\) Image Docker basée sur Windows : dotnet-sdk](#)

1) image Docker de base — Windowsservercore

(les termes du contrat de licence sont disponibles à l'[adresse : https://hub.docker.com/_/microsoft-windows-servercore](https://hub.docker.com/_/microsoft-windows-servercore))

Licence : Par la demande et l'utilisation de cette image de conteneur du système d'exploitation pour les conteneurs Windows, vous reconnaissez, comprenez et acceptez les conditions du contrat de licence complémentaire suivantes :

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

IMAGE DE CONTENEUR DU SYSTÈME D'EXPLOITATION

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) (référé en tant que « nous » ou « Microsoft ») vous accorde la licence complémentaire de cette image de conteneur du système d'exploitation (ci-après désigné « Complément »). Vous êtes autorisé à utiliser ce Complément conjointement avec le système d'exploitation hôte sous-jacent (ci-après désigné « Logiciel Hôte ») uniquement pour faciliter l'exécution des conteneurs du Logiciel Hôte. Les conditions du contrat de licence du Logiciel Hôte s'appliquent à votre utilisation du Complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le Logiciel Hôte. Vous pouvez utiliser ce Complément avec chaque copie sous licence valide du Logiciel Hôte.

AUTRES EXIGENCES ET/OU DROITS D'UTILISATION DE LA LICENCE

Votre utilisation du Complément telle que spécifiée dans le paragraphe précédent peut entraîner la création ou la modification d'une image de conteneur (ci-après désigné « Image de conteneur ») qui inclut certains composants du Complément. Pour plus de clarté, une Image de conteneur est séparée et distincte d'une machine virtuelle ou d'une image d'appliance virtuelle. Conformément aux présentes conditions du contrat de licence, nous vous accordons un droit restreint de redistribuer lesdits composants du Complément sous les conditions suivantes :

- (i) vous pouvez utiliser les composants du Complément uniquement tels qu'ils sont utilisés dans, et dans le cadre de votre Image de conteneur,
- (ii) vous pouvez utiliser lesdits composants du Complément dans votre Image de conteneur tant que vous disposez de fonctionnalités principales importantes dans votre Image de conteneur qui sont matériellement séparées et distinctes du Complément ; et
- (iii) vous acceptez d'inclure les présentes conditions du contrat de licence (ou des conditions semblables requises par nous ou un hôte) à votre Image de conteneur pour convenablement autoriser l'utilisation possible des composants du Complément par vos utilisateurs finaux.

Nous nous réservons tous les autres droits non accordés explicitement dans les présentes.

En utilisant le présent Complément, vous acceptez les présentes conditions. Si vous ne les acceptez pas, n'utilisez pas le présent Complément.

Dans le cadre des conditions du contrat de licence complémentaire pour la présente image de conteneur du système d'exploitation pour les conteneurs Windows, vous êtes également soumis aux conditions du contrat de licence du logiciel hôte Windows Server sous-jacent, qui sont disponibles à l'adresse : <https://www.microsoft.com/en-us/usetterms>.

2) Image Docker basée sur Windows : choco

(les termes du contrat de licence sont disponibles à l'[adresse suivante : https://github.com/chocolatey/choco/blob/master/LICENSE](https://github.com/chocolatey/choco/blob/master/LICENSE))

Tous droits réservés 2011 - Present RealDimensions Software, LLC

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2.0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

3) Image Docker basée sur Windows : git --version 2.16.2

(conditions du contrat de licence disponibles à l'adresse : <https://chocolatey.org/packages/git/2.16.2>)

Sous licence de GNU General Public License, version 2, disponible à l'adresse : <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

4) Image Docker basée sur Windows — --version 15.0.26320.2 microsoft-build-tools

(conditions du contrat de licence disponibles à l'adresse : <https://www.visualstudio.com/license-terms/mt171552/>)

MICROSOFT VISUAL STUDIO 2015 EXTENSIONS, VISUAL STUDIO SHELLS et C++
REDISTRIBUTABLE

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Elles s'appliquent au logiciel désigné ci-dessus.

Les conditions s'appliquent également à tous les services ou mises à jour Microsoft du logiciel, sauf dans la mesure où elles disposent de conditions supplémentaires.

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS DISPOSEZ DES DROITS CI-DESSOUS.

1. INSTALLATION ET DROITS D'UTILISATION. Vous pouvez installer et utiliser n'importe quel nombre de copies du logiciel.
2. CONDITIONS POUR LES COMPOSANTS SPÉCIFIQUES.
 - a. Utilitaires. Le logiciel peut contenir certains éléments de la liste des utilitaires à l'[adresse https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs](https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs). Vous pouvez copier et installer ces éléments, s'ils sont inclus dans le logiciel, sur les vôtres ou sur d'autres machines tierces, pour déboguer et déployer les applications et les bases de données que vous avez développées avec le logiciel. Veuillez noter que les Utilitaires sont conçus pour une utilisation temporaire, que Microsoft ne peut pas corriger ou mettre à jour les Utilitaires séparément du reste du logiciel, et que certains Utilitaires par nature peuvent permettre à d'autres d'accéder à des machines sur lesquelles ils sont installés. Par conséquent, vous devez supprimer tous les Utilitaires installés après la fin du débogage ou du déploiement de vos applications et bases de données. Microsoft n'est pas responsable de toute utilisation tierce ou de tout accès tiers à des Utilitaires installés sur n'importe quelle machine.
 - b. Plateformes Microsoft. Le logiciel peut inclure des composants provenant de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office et Microsoft SharePoint. Lesdits composants sont soumis à des accords séparés et leurs propres politiques d'assistance produit, tel que décrit dans les conditions du contrat de licence disponibles dans le répertoire d'installation dudit composant ou dans le dossier « Licences » accompagnant le logiciel.
 - c. Composants tiers. Le logiciel peut inclure des composants tiers soumis à des mentions légales distinctes ou régis par d'autres accords, comme décrit dans le ThirdPartyNotices fichier accompagnant le logiciel. Même si lesdits composants sont soumis à d'autres accords, les exclusions de responsabilités, les limitations et les exclusions concernant les dommages ci-dessous s'appliquent également. Le logiciel peut également inclure des composants sous licence de licences open source avec les obligations de disponibilité du code source. Des copies de ces licences, le cas échéant, sont incluses dans le ThirdPartyNotices fichier. Vous pouvez obtenir ledit code source grâce à nous, si nécessaire et requis par les licences open source applicables, en envoyant un mandat ou un chèque de 5 dollars américain à l'adresse :

Source Code Compliance Team, Microsoft Corporation, 1 Microsoft Way, Redmond WA 98052, États-Unis. Veuillez écrire le code source d'un ou de plusieurs des composants répertoriés ci-dessous dans la ligne réservée aux notes de votre paiement :

- Remote Tools for Visual Studio 2015 ;
- Standalone Profiler for Visual Studio 2015 ;
- IntelliTraceCollector pour Visual Studio 2015 ;
- Microsoft VC++ Redistributable 2015 ;
- Multibyte MFC Library for Visual Studio 2015 ;
- Microsoft Build Tools 2015 ;
- Feedback Client ;
- Visual Studio 2015 Integrated Shell ; ou
- Visual Studio 2015 Isolated Shell.

Nous pouvons également effectuer une copie du code source disponible à l'adresse <http://thirdpartysource.microsoft.com>.

3. **DONNÉES.** Le logiciel peut récupérer des informations sur vous et votre utilisation du logiciel, et les envoyer à Microsoft. Microsoft peut utiliser ces informations pour fournir des services et améliorer nos produits et services. Vous pouvez vous désinscrire de la plupart desdits scénarios, mais pas de tous, tel que décrit dans la documentation du produit. Il existe également des fonctionnalités dans le logiciel qui peuvent vous permettre de récupérer des données d'utilisateurs de vos applications. Si vous utilisez ces fonctionnalités pour activer la récupération de données dans vos applications, vous devez respecter la loi en vigueur, notamment en avertissant de manière appropriée les utilisateurs de vos applications. Vous pouvez en savoir plus sur la récupération et l'utilisation des données dans la documentation d'aide et la déclaration de confidentialité à l'adresse <https://privacy.microsoft.com/en-us/privacystatement>. Votre utilisation du logiciel sera considérée comme un consentement à ces pratiques.
4. **CHAMP D'APPLICATION DE LA LICENCE.** Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Vous ne pouvez pas
 - contourner n'importe quelles limitations du logiciel ;

- procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, ou essayez de le faire, sauf si exigé et uniquement dans la mesure requise par les conditions du contrat de licence tierces régissant l'utilisation de certains composants open source pouvant être inclus avec le logiciel ;
 - supprimer, minimiser, bloquer ou modifier toutes les mentions de Microsoft ou de ses fournisseurs dans le logiciel ;
 - utiliser le logiciel de manière illégale ; ou
 - partager, publier, louer ou céder en location le logiciel, ou proposer le logiciel en tant que solution hébergée autonome pour que d'autres personnes l'utilisent.
5. **RESTRICTIONS À L'EXPORTATION.** Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel, qui incluent des restrictions sur des destinations, des utilisateurs finaux, et des utilisations finales. Pour plus d'informations sur les restrictions à l'exportation, visitez (aka.ms/exporting).
6. **SERVICES D'ASSISTANCE.** Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
7. **ACCORD ENTIER.** Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et les services d'assistance.
8. **LOI APPLICABLE.** Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington s'appliquent à l'interprétation des réclamations et aux réclamations pour violation du présent accord, et les lois de l'État où vous résidez s'appliquent à toutes les autres réclamations. Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois du pays s'appliquent.
9. **DROITS DES CONSOMMATEURS ; DIFFÉRENCES RÉGIONALES.** Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits, notamment les droits des consommateurs, conformément aux lois de votre État ou pays. À part et en dehors de votre relation avec Microsoft, vous pouvez également disposer de droits concernant la partie grâce à laquelle vous avez acquis le logiciel. Cet accord ne modifie pas lesdits autres droits si les lois de votre État ou pays ne permettent pas à le faire. Par exemple, si vous avez acquis le logiciel dans l'une des régions ci-dessous, ou que la loi obligatoire du pays s'applique, les dispositions suivantes s'appliquent à vous :
- a. **Australie.** Vous disposez de garanties statutaires conformément à l'Australian Consumer Law et rien dans le présent accord n'est destiné à affecter lesdits droits.
 - b. **Canada.** Si vous avez acquis le présent logiciel au Canada, vous pouvez cesser de recevoir des mises à jour en désactivant la fonctionnalité de mise à jour automatique, déconnecter votre

appareil d'Internet (toutefois, si et quand vous le reconnectez à Internet, le logiciel reprend la vérification et l'installation des mises à jour) ou désinstaller le logiciel. La documentation produit, le cas échéant, peut également spécifier la façon de désactiver les mises à jour d'un appareil ou logiciel spécifique.

c. Allemagne et Autriche.

- i. Garantie. Le logiciel sous licence valide logiciel fonctionnera tel que décrit dans n'importe quel matériel Microsoft accompagnant le logiciel. Toutefois, Microsoft n'offre aucune garantie contractuelle concernant le logiciel sous licence.
- ii. Limitation de responsabilité. En cas de comportements intentionnels, négligence grave, réclamations reposant sur le Product Liability Act, ainsi qu'en cas de mort ou de blessures personnelles ou physiques, Microsoft sera tenu responsable conformément à la loi statutaire. Sous condition de la clause précédente (ii), Microsoft sera uniquement tenu responsable en cas de négligence mineure si Microsoft ne respecte pas lesdites obligations contractuelles matérielles, dont l'accomplissement faciliterait l'exécution parfaite du présent accord, dont le non-respect pourrait menacer l'objet même du présent accord et dont la conformité avec une partie peut constamment renforcer la confiance (ci-après désigné « Obligations cardinales »). Dans d'autres cas de négligence mineure, Microsoft ne sera pas tenu responsable.

10 EXCLUSION DE GARANTIE. LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

11 LIMITATION ET EXCLUSION DES DOMMAGES. MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES. La présente limitation s'applique (a) à tout aspect lié au logiciel, aux services (notamment le code) sur les sites Internet tiers, ou les applications tierces ; et (b) aux revendications pour violation du contrat, violation de garantie, garantie ou condition, responsabilité stricte, négligence, ou tout autre délit dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

ID EULA : ShellsRedist VS2015_Update3_ _ <ENU>

5) Image Docker basée sur Windows : nuget.commandline --version 4.5.1

(les termes du contrat de licence sont disponibles à l'[adresse suivante : https://github.com/ NuGet / Home/blob/dev/LICENSE.txt](https://github.com/NuGet/Home/blob/dev/LICENSE.txt))

Copyright (c) .NET Foundation. Tous droits réservés.

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2.0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

7) Image Docker basée sur Windows : netfx-4.6.2-devpack

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

.NET FRAMEWORK ET LES PACKAGES DE LANGUE ASSOCIÉS POUR LE SYSTÈME D'EXPLOITATION MICROSOFT WINDOWS

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) vous accorde la licence du présent complément. Si vous êtes sous licence pour utiliser le logiciel du système d'exploitation Microsoft Windows (ci-après désigné le « logiciel »), vous pouvez utiliser le présent complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le logiciel. Vous pouvez utiliser le présent Complément avec chaque copie sous licence valide du logiciel.

Les conditions du contrat de licence suivantes décrivent les conditions d'utilisation supplémentaires du présent complément. Les présentes conditions et conditions du contrat de licence du logiciel

s'appliquent à votre utilisation du complément. En cas de conflit, les présentes conditions du contrat de licence complémentaire s'appliquent.

EN UTILISANT CE SUPPLÉMENT, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES ACCEPTEZ PAS, N'UTILISEZ PAS CE SUPPLÉMENT.

Si vous respectez les présentes conditions du contrat de licence, vous disposez des droits ci-dessous.

1. CODE DISTRIBUABLE. Le supplément est composé d'un code distribuable. Un « Code Distribuable » est un code que vous êtes autorisé à distribuer dans les programmes que vous développez si vous respectez les conditions ci-dessous.
 - a. Droit d'utilisation et de distribution.
 - Vous pouvez copier et distribuer le complément sous forme de code objet.
 - Distribution tierce. Vous pouvez autoriser les distributeurs de vos programmes à copier et à distribuer le Code Distribuable dans le cadre de ces programmes.
 - b. Exigences relatives à la distribution. Pour tout code distribuable que vous distribuez, vous devez
 - lui ajouter des fonctionnalités principales importantes dans vos programmes ;
 - pour n'importe quel Code Distribuable disposant d'une extension de fichier .lib, distribuer uniquement les résultats de l'exécution dudit Code Distribuable via un éditeur de liens avec votre programme ;
 - distribuer le Code Distribuable inclus dans un programme d'installation uniquement dans le cadre de ce programme d'installation sans modification ;
 - exiger que les distributeurs et les utilisateurs finaux externes acceptent les conditions qui le protègent au moins autant que le présent accord.
 - afficher vos mentions des droits d'auteur valides sur vos programmes ; et
 - indemniser, défendre et exonérer Microsoft face à toutes les revendications, y compris les frais d'avocat, liés à la distribution ou à l'utilisation de vos programmes.
 - c. Restrictions de distribution. Il se peut que vous ne
 - modifier les droits d'auteur, les marques déposées ou les brevets du Code Distribuable ;
 - utiliser des marques déposées de Microsoft dans vos noms de programme ou en laissant croire que vos programmes proviennent de Microsoft ou sont approuvés par Microsoft ;
 - distribuer un Code Distribuable à exécuter sur une plateforme autre que Windows ;

- inclure un Code Distribuable dans des programmes malveillants, frauduleux ou illicites ; ou
- modifier ou distribuer le code source de n'importe quel Code Distribuable afin que toute partie de celui-ci soit soumise à une licence d'exclusion. Une licence d'exclusion exige, comme condition d'utilisation, modification ou distribution, que
 - le code divulgué ou distribué sous forme de code source ; ou
 - d'autres aient le droit de la modifier.

2. SERVICES D'ASSISTANCE POUR LE COMPLÉMENT. Microsoft fournit des services d'assistance pour le présent logiciel tel que décrit à l'adresse www.support.microsoft.com/common/international.aspx.

8) Image Docker basée sur Windows — visualfsharptools, version 4.0

(les termes du contrat de licence sont disponibles à l'[adresse suivante : https://github.com/dotnet/fsharp/blob/main/License.txt](https://github.com/dotnet/fsharp/blob/main/License.txt))

Copyright (c) Microsoft Corporation. Tous droits réservés.

Sous licence d'Apache License, version 2.0 (ci-après désigné « Licence ») ; vous ne pouvez pas utiliser les présents fichiers sauf conformément à la Licence. Vous pouvez obtenir une copie de la Licence à l'adresse

<http://www.apache.org/licenses/LICENSE-2.0>

Sauf si la loi applicable l'exige ou sauf accord écrit, le logiciel distribué conformément à la Licence est distribué « EN L'ÉTAT », SANS GARANTIE OU CONDITION D'AUCUNE SORTE, explicite ou implicite. Consultez la Licence pour les dispositions spécifiques régissant les autorisations et limitations dans le cadre de la Licence.

9) Image Docker basée sur Windows : -4,6 netfx-pcl-reference-assemblies

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE MICROSOFT

ASSEMBLAGES DE RÉFÉRENCE DE LA BIBLIOTHÈQUE DE CLASSE PORTABLE
MICROSOFT .NET – 4.6

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Veuillez les lire. Elles s'appliquent au logiciel désigné ci-dessus. Les conditions s'appliquent également à l'ensemble des

- mises à jour,
- compléments,
- services Internet, et
- services d'assistance Microsoft

pour ce logiciel, sauf si d'autres conditions accompagnent lesdits éléments. Si tel est le cas, lesdites conditions s'appliquent.

EN UTILISANT LE LOGICIEL, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES ACCEPTEZ PAS, N'UTILISEZ PAS LE LOGICIEL.

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS DISPOSEZ DES DROITS PERPÉTUELS CI-DESSOUS.

1. INSTALLATION ET DROITS D'UTILISATION. Vous pouvez installer et utiliser n'importe quel nombre de copies du logiciel pour concevoir, développer et tester vos programmes.
2. AUTRES EXIGENCES ET/OU DROITS D'UTILISATION DE LA LICENCE.
 - a. Code Distribuible. Vous pouvez distribuer le logiciel dans les programmes d'outils de développement que vous développez, pour permettre aux clients de vos programmes de développer des bibliothèques portables à utiliser avec n'importe quel appareil ou système d'exploitation, si vous respectez les conditions ci-dessous.
 - i. Droit d'utilisation et de distribution. Le logiciel s'appelle « Code distribuible ».
 - Code Distribuible. Vous pouvez copier et distribuer le logiciel sous forme de code objet.
 - Distribution tierce. Vous pouvez autoriser les distributeurs de vos programmes à copier et à distribuer le Code Distribuible dans le cadre de ces programmes.
 - ii. Exigences relatives à la distribution. Pour tout code distribuible que vous distribuez, vous devez
 - lui ajouter des fonctionnalités principales importantes dans vos programmes ;

- exiger que les distributeurs et vos clients acceptent les conditions qui le protègent au moins autant que le présent accord.
- afficher vos mentions des droits d'auteur valides sur vos programmes ; et
- indemniser, défendre et exonérer Microsoft face à toutes les revendications, y compris les frais d'avocat, liés à la distribution ou à l'utilisation de vos programmes.

iii. Restrictions de distribution. Il se peut que vous ne

- modifier les droits d'auteur, les marques déposées ou les brevets du Code Distribuable ;
- utiliser des marques déposées de Microsoft dans vos noms de programme ou en laissant croire que vos programmes proviennent de Microsoft ou sont approuvés par Microsoft ;
- inclure un Code Distribuable dans des programmes malveillants, frauduleux ou illicites ; ou
- modifier ou distribuer le Code Distribuable afin que toute partie de celui-ci soit soumise à une licence d'exclusion. Une licence d'exclusion exige, comme condition d'utilisation, modification ou distribution, que
 - le code divulgué ou distribué sous forme de code source ; ou
 - d'autres aient le droit de la modifier.

3. CHAMP D'APPLICATION DE LA LICENCE. Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Vous ne pouvez pas

- contourner n'importe quelles limitations du logiciel ;
- procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, sauf si exigé et uniquement dans la mesure autorisée expressément par la législation applicable, malgré la présente limitation ;
- publier le logiciel pour que d'autres le copient ; ou
- louer, céder en location ou prêter le logiciel.

4. COMMENTAIRES. Vous pouvez fournir des commentaires sur le logiciel. Si vous communiquer des commentaires sur le logiciel à Microsoft, vous accordez à Microsoft, sans frais, le droit d'utiliser, de partager et de commercialiser vos commentaires de quelque façon que ce soit et à quelque fin que ce soit. Vous accordez également à des tiers, sans frais, les droits de brevet nécessaires à leurs produits, technologies et services afin d'utiliser ou de communiquer avec n'importe quelle partie spécifique d'un logiciel ou d'un service Microsoft qui inclut les

- commentaires. Vous ne pourrez pas communiquer de commentaires soumis à une licence qui exige que Microsoft concède des licences pour son logiciel ou sa documentation à un tiers, car vos commentaires y sont inclus. Ces droits survivent au présent accord.
5. **TRANSFERT À UN TIERS.** Le premier utilisateur du logiciel peut le transférer, ainsi que le présent accord, directement à un tiers. Avant le transfert, ce tiers doit accepter que le présent accord s'applique au transfert et à l'utilisation du logiciel. Le premier utilisateur doit désinstaller le logiciel avant de le transférer séparément depuis l'appareil. Le premier utilisateur ne peut conserver aucune copie.
 6. **RESTRICTIONS À L'EXPORTATION.** Le logiciel est soumis lois et réglementations d'exportation américaines. Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel. Lesdites lois incluent des restrictions sur des destinations, des utilisateurs finaux, et des utilisations finales. Pour plus d'informations, consultez www.microsoft.com/exporting.
 7. **SERVICES D'ASSISTANCE.** Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
 8. **ACCORD ENTIER.** Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et tous les services d'assistance que nous fournissons.
 9. **LOI APPLICABLE.**
 - a. **États-Unis.** Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington régissent l'interprétation du présent accord et s'appliquent aux réclamations pour violation de ce dernier, indépendamment de toute règle de conflit de lois. Les lois de l'État dans lequel vous résidez régissent toutes les autres réclamations, notamment les réclamations en vertu des lois sur la protection des consommateurs, des lois sur la concurrence déloyale, et des lois sur la responsabilité civile.
 - b. **En dehors des États-Unis.** Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois de ce dernier s'appliquent.
 10. **CONSÉQUENCE LÉGALE.** Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits en vertu des lois de votre pays. Vous pouvez également disposer de droits liés au tiers grâce auquel vous avez acquis le logiciel. Le présent accord ne modifie pas vos droits en vertu des lois de votre pays si les lois de votre pays ne permettent pas à le faire.
 11. **EXCLUSION DE GARANTIE. LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. VOUS POUVEZ DISPOSER DE DROITS DES CONSOMMATEURS SUPPLÉMENTAIRES OU DE GARANTIES STATUTAIRE EN VERTU**

DE VOS LOIS LOCALES QUE LE PRÉSENT ACCORD NE PEUT PAS MODIFIER. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

POUR L'AUSTRALIE—VOUS DISPOSEZ DE GARANTIES STATUTAIRES CONFORMÉMENT À L'AUSTRALIAN CONSUMER LAW ET RIEN DANS LE PRÉSENT ACCORD N'EST DESTINÉ À AFFECTER LESDITS DROITS.

12LIMITATION ET EXCLUSION DES RECOURS ET DES DOMMAGES-INTÉRÊTS. MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES.

Cette limitation s'applique

- à tout ce qui concerne le logiciel, les services, le contenu (notamment le code) sur les sites Internet tiers, ou les programmes tiers ; et
- aux réclamations pour violation de contrat, violation de garantie ou de condition, responsabilité stricte, négligence, ou autres délits dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

10) Image Docker basée sur Windows — visualcppbuildtools v 14.0.25420.1

(conditions du contrat de licence disponibles à l'adresse : <https://www.visualstudio.com/license-terms/mt644918/>)

OUTILS DE GÉNÉRATION MICROSOFT VISUAL C++

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE MICROSOFT

OUTILS DE GÉNÉRATION MICROSOFT VISUAL C++

Les présentes conditions du contrat de licence sont un accord entre Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) et vous. Elles s'appliquent au logiciel désigné ci-dessus. Les conditions s'appliquent également à tous les services ou mises à jour Microsoft du logiciel, sauf dans la mesure où elles disposent de conditions différentes.

SI VOUS RESPECTEZ LES PRÉSENTES CONDITIONS DU CONTRAT DE LICENCE, VOUS DISPOSEZ DES DROITS CI-DESSOUS.

1. INSTALLATION ET DROITS D'UTILISATION.

a. Un utilisateur peut utiliser des copies du logiciel pour développer et tester ses applications.

2. DONNÉES. Le logiciel peut récupérer des informations sur vous et votre utilisation du logiciel, et les envoyer à Microsoft. Microsoft peut utiliser ces informations pour fournir des services et améliorer nos produits et services. Vous pouvez vous désinscrire de la plupart desdits scénarios, mais pas de tous, tel que décrit dans la documentation du produit. Il existe également des fonctionnalités dans le logiciel qui peuvent vous permettre de récupérer des données d'utilisateurs de vos applications. Si vous utilisez ces fonctionnalités pour activer la récupération de données dans vos applications, vous devez respecter la loi en vigueur, notamment en avertissant de manière appropriée les utilisateurs de vos applications. Vous pouvez en savoir plus sur la récupération et l'utilisation des données dans la documentation d'aide et la déclaration de confidentialité à l'adresse <http://go.microsoft.com/fwlink/?LinkID=528096>. Votre utilisation du logiciel sera considérée comme un consentement à ces pratiques.

3. CONDITIONS POUR LES COMPOSANTS SPÉCIFIQUES.

a. Build Server. Le logiciel peut contenir certains composants du serveur de compilation répertoriés dans des fichiers BuildServer .TXT et/ou des fichiers répertoriés dans la BuildServer liste située conformément aux présentes conditions de licence logicielle Microsoft. Vous pouvez copier et installer ces éléments, s'ils sont inclus dans le logiciel, sur vos machines de génération. Vous et d'autres utilisateurs de votre organisation peuvent utiliser ces éléments sur vos machines de génération uniquement à des fins de compilation, de génération, de vérification et d'archivage de vos applications ou d'exécution de tests de qualité ou de performance dans le cadre du processus de génération.

b. Plateformes Microsoft. Le logiciel peut inclure des composants provenant de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office et Microsoft SharePoint. Lesdits composants sont soumis à des accords séparés et leurs propres politiques d'assistance produit, tel que décrit dans les conditions du contrat de licence

disponibles dans le répertoire d'installation dudit composant ou dans le dossier « Licences » accompagnant le logiciel.

- c. Composants tiers. Le logiciel peut inclure des composants tiers soumis à des mentions légales distinctes ou régis par d'autres accords, comme décrit dans le ThirdPartyNotices fichier accompagnant le logiciel. Même si lesdits composants sont soumis à d'autres accords, les exclusions de responsabilités, les limitations et les exclusions concernant les dommages ci-dessous s'appliquent également.
 - d. Gestionnaires de package. Le logiciel peut inclure des gestionnaires de package, tels que Nuget, qui vous permettent de télécharger d'autres packages logiciels Microsoft et des packages logiciels tiers à utiliser avec votre application. Ces packages sont soumis à leurs propres licences, et non pas au présent accord. Microsoft ne permet pas de distribuer, d'autoriser ou de fournir des garanties pour l'un des packages tiers.
4. CHAMP D'APPLICATION DE LA LICENCE. Le logiciel est fourni sous licence et non vendu. Cet accord vous concède uniquement certains droits pour utiliser le logiciel. Microsoft se réserve tous les autres droits. Sauf si la loi applicable vous concède davantage de droits malgré cette limitation, vous pouvez utiliser le logiciel uniquement tel qu'expressément autorisé dans ledit accord. Pour ce faire, vous devez respecter toutes les limitations techniques du logiciel qui vous autorisent à l'utiliser d'une certaine façon. Pour plus d'informations, consultez <https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms#1-installation-and-use-rights>. Vous ne pouvez pas
- contourner n'importe quelles limitations du logiciel ;
 - procéder à des opérations de rétro-ingénierie, décompiler ou désassembler le logiciel, ou essayez de le faire, sauf si exigé et uniquement dans la mesure requise par les conditions du contrat de licence tierces régissant l'utilisation de certains composants open source pouvant être inclus avec le logiciel ;
 - supprimer, minimiser, bloquer ou modifier toutes les mentions de Microsoft ou de ses fournisseurs ;
 - utiliser le logiciel de manière illégale ; ou
 - partager, publier, louer ou céder en location le logiciel, ou proposer le logiciel en tant que solution hébergée autonome pour que d'autres personnes l'utilisent.
5. RESTRICTIONS À L'EXPORTATION. Vous devez respecter toutes les lois et réglementations d'exportation nationales et internationales qui s'appliquent au logiciel, qui incluent des restrictions sur des destinations, des utilisateurs finaux et des utilisations finales. Pour plus d'informations sur les restrictions à l'exportation, visitez (aka.ms/exporting).

6. SERVICES D'ASSISTANCE. Étant donné que ce logiciel est fourni « en l'état », nous ne pouvons pas fournir de services d'assistance.
7. ACCORD ENTIER. Cet accord, et les conditions pour les compléments, les mises à jour, les services Internet et les services d'assistance que vous utilisez, représentent l'accord entier pour le logiciel et les services d'assistance.
8. LOI APPLICABLE. Si vous avez acquis le logiciel aux États-Unis, les lois de l'État de Washington s'appliquent à l'interprétation des réclamations et aux réclamations pour violation du présent accord, et les lois de l'État où vous résidez s'appliquent à toutes les autres réclamations. Si vous avez acquis le logiciel dans n'importe quel autre pays, les lois du pays s'appliquent.
9. DROITS DES CONSOMMATEURS ; DIFFÉRENCES RÉGIONALES. Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits, notamment les droits des consommateurs, conformément aux lois de votre État ou pays. À part et en dehors de votre relation avec Microsoft, vous pouvez également disposer de droits concernant la partie grâce à laquelle vous avez acquis le logiciel. Cet accord ne modifie pas lesdits autres droits si les lois de votre État ou pays ne permettent pas à le faire. Par exemple, si vous avez acquis le logiciel dans l'une des régions ci-dessous, ou que la loi obligatoire du pays s'applique, les dispositions suivantes s'appliquent à vous :
 - Australie. Vous disposez de garanties statutaires conformément à l'Australian Consumer Law et rien dans le présent accord n'est destiné à affecter lesdits droits.
 - Canada. Si vous avez acquis le présent logiciel au Canada, vous pouvez cesser de recevoir des mises à jour en désactivant la fonctionnalité de mise à jour automatique, déconnecter votre appareil d'Internet (toutefois, si et quand vous le reconnectez à Internet, le logiciel reprend la vérification et l'installation des mises à jour) ou désinstaller le logiciel. La documentation produit, le cas échéant, peut également spécifier la façon de désactiver les mises à jour d'un appareil ou logiciel spécifique.
 - Allemagne et Autriche.
 - Garantie. Le logiciel sous licence valide fonctionnera tel que décrit dans n'importe quel matériel Microsoft accompagnant le logiciel. Toutefois, Microsoft n'offre aucune garantie contractuelle concernant le logiciel sous licence.
 - Limitation de responsabilité. En cas de comportements intentionnels, négligence grave, réclamations reposant sur le Product Liability Act, ainsi qu'en cas de mort ou de blessures personnelles ou physiques, Microsoft sera tenu responsable conformément à la loi statutaire.

Sous condition de la clause précédente (ii), Microsoft sera uniquement tenu responsable en cas de négligence mineure si Microsoft ne respecte pas lesdites obligations contractuelles

matérielles, dont l'accomplissement faciliterait l'exécution parfaite du présent accord, dont le non-respect pourrait menacer l'objet même du présent accord et dont la conformité avec une partie peut constamment renforcer la confiance (ci-après désigné « Obligations cardinales »). Dans d'autres cas de négligence mineure, Microsoft ne sera pas tenu responsable.

10. CONSÉQUENCE LÉGALE. Cet accord décrit certains droits juridiques. Vous pouvez disposer d'autres droits en vertu des lois de votre État ou pays. Le présent accord ne modifie pas vos droits en vertu des lois de votre État ou pays si les lois de votre État ou pays ne permettent pas à le faire. Sans restreindre la portée générale de ce qui précède, pour l'Australie, VOUS DISPOSEZ DE GARANTIES STATUTAIRES CONFORMÉMENT À L'AUSTRALIAN CONSUMER LAW ET RIEN DANS LE PRÉSENT ACCORD N'EST DESTINÉ À AFFECTER LESDITS DROITS

11. EXCLUSION DE GARANTIE. LE LOGICIEL EST FOURNI SOUS LICENCE « EN L'ÉTAT ». VOUS ASSUMEZ LE RISQUE DE L'UTILISER. MICROSOFT N'ACCORDE AUCUNE GARANTIE EXPRESSE OU CONDITION. DANS LES LIMITES AUTORISÉES PAR LES LOIS LOCALES, MICROSOFT EXCLUT LES GARANTIES IMPLICITES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET DE NON-CONTREFAÇON.

12. LIMITATION ET EXCLUSION DES DOMMAGES. MICROSOFT ET VOUS POUVEZ RECOVER SES FOURNISSEURS ONLY DOMMAGES DIRECTS JUSQU'À U.S. 5,00 USD. VOUS NE POUVEZ PAS OBTENIR RÉPARATION POUR TOUT AUTRE DOMMAGE, Y COMPRIS LES DOMMAGES CONSÉCUTIFS, LES PERTES DE PROFITS, LES DOMMAGES SPÉCIAUX, INDIRECTS OU ACCESSOIRES.

La présente limitation s'applique (a) à tout aspect lié au logiciel, aux services (notamment le code) sur les sites Internet tiers, ou les applications tierces ; et (b) aux revendications pour violation du contrat, violation de garantie, garantie ou condition, responsabilité stricte, négligence, ou tout autre délit dans les limites autorisées par la loi applicable.

Elle s'applique également, même si Microsoft avait connaissance ou aurait dû avoir connaissance de la possibilité de dommages. La limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous, car votre pays peut ne pas autoriser l'exclusion ou la limitation des dommages fortuits, punitifs ou autres.

11) Image Docker basée sur Windows : 3-ondemand-package.cab microsoft-windows-netfx

CONDITIONS DU CONTRAT DE LICENCE COMPLÉMENTAIRE DU LOGICIEL MICROSOFT

MICROSOFT .NET FRAMEWORK 3.5 SP1 POUR SYSTÈME D'EXPLOITATION MICROSOFT WINDOWS

Microsoft Corporation (ou selon votre lieu de résidence, l'une de ses filiales) vous accorde la licence du présent complément. Si vous êtes sous licence pour utiliser le logiciel du système d'exploitation Microsoft Windows (pour lequel le présent complément s'applique) (ci-après désigné le « logiciel »), vous pouvez utiliser le présent complément. Vous ne pouvez pas l'utiliser si vous ne disposez pas d'une licence pour le logiciel. Vous pouvez utiliser une copie du présent Complément avec chaque copie sous licence valide du logiciel.

Les conditions du contrat de licence suivantes décrivent les conditions d'utilisation supplémentaires du présent complément. Les présentes conditions et conditions du contrat de licence du logiciel s'appliquent à votre utilisation du complément. En cas de conflit, les présentes conditions du contrat de licence complémentaire s'appliquent.

EN UTILISANT CE SUPPLÉMENT, VOUS ACCEPTEZ CES CONDITIONS. SI VOUS NE LES ACCEPTEZ PAS, N'UTILISEZ PAS CE SUPPLÉMENT.

Si vous respectez les présentes conditions du contrat de licence, vous disposez des droits ci-dessous.

1. SERVICES D'ASSISTANCE POUR LE COMPLÉMENT. Microsoft fournit des services d'assistance pour le présent logiciel tel que décrit à l'adresse www.support.microsoft.com/common/international.aspx.
2. TESTS D'ÉVALUATION MICROSOFT .NET. Le logiciel inclut .NET Framework, Windows Communication Foundation, Windows Presentation Foundation, et les composants Windows Workflow Foundation des systèmes d'exploitation Windows (.NET Components). Vous pouvez effectuer des tests d'évaluation internes de .NET Components. Vous pouvez communiquer les résultats de n'importe quel test d'évaluation des .NET Components, à condition que vous respectiez les conditions stipulées à l'adresse <http://go.microsoft.com/fwlink/?LinkID=66406>.

Nonobstant tout autre accord avec Microsoft, si vous communiquez les résultats des tests d'évaluation, Microsoft se réserve le droit de communiquer les résultats des tests d'évaluation qu'il réalise pour vos produits qui concurrencent le .NET Component applicable, pourvu qu'il respecte les mêmes conditions stipulées à l'adresse <http://go.microsoft.com/fwlink/?LinkID=66406>.

12) Image Docker basée sur Windows : dotnet-sdk

(disponible sur <https://github.com/dotnet/core/blob/main/LICENSE.TXT>)

La licence MIT (MIT)

Copyright (c) Microsoft Corporation

Par la présente, une permission est accordée, sans frais, à toute personne obtenant une copie du présent logiciel et des fichiers de documentation associés (ci-après désigné le « Logiciel »), pour utiliser le Logiciel sans restriction, y compris mais sans s'y limiter les droits d'utiliser, de copier, de modifier, de fusionner, de publier, de distribuer, de concéder en sous-licence, et/ou de vendre des copies du Logiciel, et de permettre à des personnes à qui le Logiciel est fourni de faire de même, en étant soumise aux conditions suivantes :

Les mentions des droits d'auteur et de ladite permission susmentionnées doivent être incluses dans toutes les copies ou parties importantes du Logiciel.

LE LOGICIEL EST FOURNI « EN L'ÉTAT », SANS GARANTIE AUCUNE, EXPRESSE OU IMPLICITE, Y COMPRIS SANS S'Y LIMITER LA GARANTIE DE QUALITE MARCHANDE, D'ADEQUATION A UN OBJECTIF PARTICULIER ET DE NON-INFRACTION. Tous droits réservés.

AWS CodeBuild Historique du document du guide de l'utilisateur

Le tableau suivant décrit les modifications importantes apportées à la documentation depuis la dernière version de AWS CodeBuild. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

- Dernière version de l'API : 2016-10-06

| Modification | Description | Date |
|---|---|-----------------|
| Nouveau contenu : ajout de nouveaux événements de webhook et de nouveaux types de filtres | Ajoutez la prise en charge des nouveaux événements webhook (RELEASED et PRERELEASED) et des nouveaux types de filtres (TAG_NAME et RELEASE_NAME). | 15 mars 2024 |
| Nouveau contenu : Ajoutez un nouvel événement webhook : PULL_REQUEST_CLOSED | Ajouter le support pour un nouvel événement webhook : PULL_REQUEST_CLOSED . | 20 février 2024 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajouter le support pour Windows Server Core 2019 (windows-base:2019-3.0) | 7 février 2024 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajout de la prise en charge des nouveaux environnements d'exécution pour Amazon Linux 2023 () a12/aarch64/standard/3.0 | 29 janvier 2024 |

| | | |
|---|---|------------------|
| Nouveau contenu : capacité réservée | CodeBuild prend désormais en charge les flottes de capacité réservée en. CodeBuild | 18 janvier 2024 |
| Nouveau type de calcul | CodeBuild supporte désormais un type de calcul Linux XLarge. Pour de plus amples informations, veuillez consulter Types de calcul de l'environnement de génération. | 8 janvier 2024 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajout de la prise en charge des nouveaux environnements d'exécution pour Amazon Linux (2a12/standard/5.0) et Ubuntu (ubuntu/standard/7.0) | 14 décembre 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajout de la prise en charge des nouvelles images de calcul Lambda | 8 décembre 2023 |
| Nouveau contenu : AWS Lambda calcul | Ajouter du nouveau contenu pour le AWS Lambda calcul | 6 novembre 2023 |
| Nouveau contenu : GitHub Action Runner | Ajouter du nouveau contenu pour l' GitHub Action Runner | 6 juillet 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajouter le support pour Amazon Linux 2 (a12/standard/5.0) | 17 mai 2023 |

| | | |
|--|--|---------------|
| Modifications apportées aux politiques gérées pour CodeBuild | Les informations relatives aux mises à jour des politiques AWS gérées pour CodeBuild sont désormais disponibles. Pour plus d'informations, voir les CodeBuild mises à jour des politiques AWS gérées . | 16 mai 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Supprimer le support pour Amazon Linux 2 (a12/standard/3.0) et ajouter le support pour Amazon Linux 2 (a12/standard/correcto8) et Amazon Linux 2 (a12/standard/correcto11) | 9 mai 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Ajouter le support pour Ubuntu 22.04 (ubuntu/standard/7.0) | 13 avril 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Supprimer le support pour Ubuntu 18.04 (ubuntu/standard/4.0) et Amazon Linux 2 (a12/aarch64/standard/1.0) | 31 mars 2023 |
| Contenu mis à jour : Images Docker fournies par CodeBuild | Supprimer le support pour Ubuntu 18.04 (ubuntu/standard/3.0) et Amazon Linux 2 (a12/standard/2.0) | 30 juin 2022 |

[Amazon ECR Sample :
Restreindre l'accès aux
images](#)

Lorsque les CodeBuild informations d'identification sont utilisées pour extraire une image Amazon ECR, vous pouvez restreindre l'accès à l'image à un CodeBuild projet spécifique. Pour plus d'informations, consultez l'[exemple Amazon ECR](#).

10 mars 2022

[Prise en charge de régions
supplémentaires](#)

Le type de ARM_CONTAINER calcul est désormais pris en charge dans les régions supplémentaires suivantes : Asie-Pacifique (Séoul), Canada (centre), Europe (Londres) et Europe (Paris). Pour de plus amples informations, veuillez consulter [Types de calcul de l'environnement de génération](#).

10 mars 2022

[Nouvelle limitation en matière
de VPC](#)

Si vous configurez CodeBuild pour fonctionner avec un VPC, la mise en cache locale n'est pas prise en charge. À compter du 28/02/22, le build de votre VPC prendra plus de temps car une nouvelle instance Amazon EC2 sera utilisée pour chaque build.

25 février 2022

| | | |
|--|---|-------------------|
| Mode de rapport par lots | CodeBuild vous permet désormais de sélectionner la manière dont les statuts de génération par lots sont envoyés au fournisseur source d'un projet. Pour plus d'informations, consultez la section Mode de rapport Batch . | 4 octobre 2021 |
| Nouveau type de calcul | CodeBuild prend désormais en charge un petit type de calcul ARM. Pour de plus amples informations, veuillez consulter Types de calcul de l'environnement de génération . | 13 septembre 2021 |
| Projets de construction publics | CodeBuild vous permet désormais de mettre les résultats de construction de vos projets de construction à la disposition du public sans avoir besoin d'accéder à un AWS compte. Pour plus d'informations, consultez la section Projets de construction publics . | 11 août 2021 |
| Débogage de session pour les compilations par lots | CodeBuild prend désormais en charge le débogage de session pour les compilations par lots. Pour plus d'informations, consultez build-graph et build-list . | 3 mars 2021 |

| | | |
|---|--|-----------------|
| Limite de construction simultanée au niveau du projet | CodeBuild vous permet désormais de limiter le nombre de builds simultanés pour un projet de build. Pour plus d'informations, voir Configuration du projet et concurrentBuildLimit . | 16 février 2021 |
| Nouvelle propriété buildspec : s3-prefix | CodeBuild fournit désormais la propriété <code>s3-prefix</code> buildspec pour les artefacts qui vous permet de spécifier un préfixe de chemin pour les artefacts chargés sur Amazon S3. Pour plus d'informations, consultez la section s3-prefix . | 9 février 2021 |
| Nouvelle propriété buildspec : en cas d'échec | CodeBuild fournit désormais la propriété <code>on-failure</code> buildspec pour les phases de construction qui vous permet de déterminer ce qui se passe en cas d'échec d'une phase de construction. Pour plus d'informations, consultez la section en cas d'échec . | 9 février 2021 |
| Nouvelle propriété buildspec : exclude-paths | CodeBuild fournit désormais la propriété <code>exclude-paths</code> buildspec pour les artefacts qui vous permet d'exclure des chemins de vos artefacts de construction. Pour plus d'informations, consultez exclude-paths . | 9 février 2021 |

| | | |
|--|--|-----------------|
| Nouvelle propriété buildspec : enable-symlinks | CodeBuild fournit désormais la propriété <code>enable-symlinks</code> buildspec pour les artefacts qui vous permet de conserver les liens symboliques dans un artefact ZIP. Pour plus d'informations, consultez la section enable-symlinks . | 9 février 2021 |
| Amélioration du nom des artefacts Buildspec | CodeBuild permet désormais à la propriété <code>artifacts/name</code> de contenir des informations de chemin. Pour plus d'informations, voir le nom . | 9 février 2021 |
| Rapports sur la couverture du code | CodeBuild fournit désormais des rapports sur la couverture du code. Pour plus d'informations, consultez les rapports de couverture du code . | 30 juillet 2020 |
| Constructions par lots | CodeBuild prend désormais en charge l'exécution de versions simultanées et coordonnées d'un projet. Pour plus d'informations, consultez la section Batch builds in CodeBuild . | 30 juillet 2020 |
| Image de Windows Server 2019 | CodeBuild fournit désormais une image de build de Windows Server Core 2019. Pour plus d'informations, consultez les images Docker fournies par CodeBuild . | 20 juillet 2020 |

| | | |
|--|---|-----------------|
| Gestionnaire de sessions | CodeBuild vous permet désormais de suspendre une construction en cours, puis d'utiliser le gestionnaire de AWS Systems Manager session pour vous connecter au conteneur de génération et afficher l'état du conteneur . Pour plus d'informations, consultez la section Gestionnaire de sessions . | 20 juillet 2020 |
| Rubrique mise à jour | CodeBuild prend désormais en charge la spécification d'un shell à utiliser dans leurs environnements de construction dans le fichier buildspec . Pour plus d'informations, consultez la section Référence des spécifications de construction . | 25 juin 2020 |
| Rapports de test avec des frameworks de test | Ajout de plusieurs rubriques décrivant comment générer des rapports de CodeBuild test avec plusieurs frameworks de test. Pour de plus amples informations, veuillez consulter Rapports de test avec des cadres de test . | 29 mai 2020 |
| Sujets mis à jour | CodeBuild prend désormais en charge l'ajout de balises aux groupes de rapports. Pour plus d'informations, consultez ReportGroup . | 21 mai 2020 |

[Support pour les rapports de test](#)

CodeBuild le support pour les rapports de test est désormais généralement disponible.

21 mai 2020

[Sujets mis à jour](#)

CodeBuild permet désormais de créer des filtres de création de webhook pour Github et Bitbucket qui déclenchent des builds uniquement lorsque le message de validation principal correspond à l'expression spécifiée. Pour plus d'informations, consultez les sections Exemple [GitHub de demande d'extraction et de filtre Webhook](#) et Exemple de [demande d'extraction et de filtre Webhook de Bitbucket](#).

6 mai 2020

[Nouveaux sujets](#)

CodeBuild prend désormais en charge le partage des ressources du projet de construction et du groupe de rapports. Pour de plus amples informations, veuillez consulter [Utilisation de projets partagés](#) et [Utilisation de groupes de rapports partagés](#).

13 décembre 2019

| | | |
|---|---|------------------|
| Sujets nouveaux et mis à jour | CodeBuild prend désormais en charge les rapports de test pendant l'exécution d'un projet de construction. Pour plus d'informations, consultez les sections Utilisation des rapports de test , Création d'un rapport de test et Création d'un rapport de test à l'aide de l'AWS CLI exemple . | 25 novembre 2019 |
| Rubrique mise à jour | CodeBuild prend désormais en charge les types d'environnement Linux GPU et Arm, ainsi que le type de 2xLarge calcul. Pour de plus amples informations, veuillez consulter Types de calcul de l'environnement de génération . | 19 novembre 2019 |
| Sujets mis à jour | CodeBuild prend désormais en charge les numéros de version sur toutes les versions, l'exportation de variables d'environnement et AWS Secrets Manager l'intégration. Pour de plus amples informations, veuillez consulter Variables exportées et Secrets Manager dans la Syntaxe des spécifications de génération . | 6 novembre 2019 |

| | | |
|-----------------------------------|---|-------------------|
| Nouvelle rubrique | CodeBuild prend désormais en charge les règles de notification. Vous pouvez utiliser des règles de notification pour informer les utilisateurs des modifications importantes apportées aux projets de génération. Pour de plus amples informations, veuillez consulter Création d'une règle de notification . | 5 novembre 2019 |
| Sujets mis à jour | CodeBuild supporte désormais les runtimes Android version 29 et Go version 1.13. Pour de plus amples informations, veuillez consulter Images Docker fournies par CodeBuild et Syntaxe des spécifications de génération . | 10 septembre 2019 |
| Sujets mis à jour | Lorsque vous créez un projet, vous pouvez désormais choisir l'image gérée Amazon Linux 2 (AL2). Pour de plus amples informations, veuillez consulter Images Docker fournies par CodeBuild et Versions d'environnement d'exécution dans l'exemple de fichier buildspec pour CodeBuild . | 16 août 2019 |

Rubrique mise à jour

Lorsque vous créez un projet, vous pouvez désormais choisir de désactiver le chiffrement des journaux S3 et, si vous utilisez un référentiel source basé sur Git, inclure des sous-modules Git. Pour plus d'informations, voir [Créer un projet de construction dans CodeBuild](#).

8 mars 2019

Nouvelle rubrique

CodeBuild prend désormais en charge la mise en cache locale. Vous pouvez spécifier la mise en cache locale dans un ou plusieurs des quatre modes lors de la création d'une génération.. Pour plus d'informations, consultez la section [Intégrer la mise en CodeBuild cache](#).

21 février 2019

Nouveaux sujets

CodeBuild prend désormais en charge les groupes de filtres Webhook pour spécifier les événements qui déclenchent une compilation. Pour plus d'informations, consultez [Filtrer les événements GitHub webhook et Filtrer les événements webhook Bitbucket](#).

8 février 2019

| | | |
|--|--|-----------------|
| Nouvelle rubrique | Le guide de CodeBuild l'utilisateur indique désormais comment utiliser CodeBuild un serveur proxy. Pour plus d'informations, consultez la section Utilisation CodeBuild avec un serveur proxy . | 4 février 2019 |
| Sujets mis à jour | CodeBuild prend désormais en charge l'utilisation d'une image Amazon ECR enregistrée dans un autre AWS compte. Plusieurs rubriques ont été mises à jour pour refléter cette modification, notamment l' exemple Amazon ECR pour CodeBuild , Create a build project et Create a CodeBuild service role . | 24 janvier 2019 |
| Support pour les registres Docker privés | CodeBuild prend désormais en charge l'utilisation d'une image Docker stockée dans un registre privé comme environnement d'exécution. Pour plus d'informations, voir Registre privé avec AWS Secrets Manager échantillon . | 24 janvier 2019 |

| | | |
|--|---|------------------|
| Rubrique mise à jour | CodeBuild prend désormais en charge l'utilisation d'un jeton d'accès pour se connecter aux référentiels GitHub (avec un jeton d'accès personnel) et Bitbucket (avec un mot de passe d'application). Pour de plus amples informations, veuillez consulter Création d'un projet de génération (console) et Utilisation des jetons d'accès avec votre fournisseur de source . | 6 décembre 2018 |
| Rubrique mise à jour | CodeBuild prend désormais en charge de nouvelles métriques de construction qui mesurent la durée de chaque phase d'une construction. Pour plus d'informations, consultez la section CodeBuild CloudWatch Mesures . | 15 novembre 2018 |
| Rubrique relative à la politique des points de terminaison VPC | Les points de terminaison Amazon VPC prennent CodeBuild actuellement en charge les politiques. Pour de plus amples informations, veuillez consulter Création d'une stratégie de point de terminaison de VPC pour CodeBuild . | 9 novembre 2018 |
| Contenu mis à jour | Les rubriques ont été mises à jour pour refléter cette nouvelle console. | 30 octobre 2018 |

[Exemple Amazon EFS](#)

CodeBuild peut monter un système de fichiers Amazon EFS lors d'une construction à l'aide des commandes du fichier buildspec d'un projet. Pour plus d'informations, consultez [l'exemple Amazon EFS pour CodeBuild](#).

26 octobre 2018

[Webhooks Bitbucket](#)

CodeBuild supporte désormais les webhooks lorsque vous utilisez Bitbucket pour votre dépôt. Pour de plus amples informations, veuillez consulter [Exemple de demande d'extraction Bitbucket pour CodeBuild](#).

2 octobre 2018

[Journaux S3](#)

CodeBuild prend désormais en charge les journaux de génération dans un compartiment S3. Auparavant, vous ne pouviez créer des journaux qu'à l'aide de CloudWatch journaux. Pour de plus amples informations, veuillez consulter [Création d'un projet](#).

17 septembre 2018

[Sources d'entrée multiples et artefacts de sortie multiples](#)

CodeBuild prend désormais en charge les projets qui utilisent plusieurs sources d'entrée et publient plusieurs ensembles d'artefacts. Pour plus d'informations, voir [Échantillonnage et CodePipeline intégration de plusieurs sources d'entrée et d'artefacts d'entrée CodeBuild et échantillon de sources d'entrée et d'artefacts de sortie multiples](#). 30 août 2018

[Exemple de versionnement sémantique](#)

Le guide de CodeBuild l'utilisateur contient désormais un exemple basé sur des cas d'utilisation qui montre comment utiliser le versionnement sémantique pour créer des noms d'artefacts au moment de la création. Pour de plus amples informations, veuillez consulter [Exemple de gestion sémantique des versions de vos artefacts de génération](#). 14 août 2018

[Nouvel exemple de site Web statique](#)

Le guide de CodeBuild l'utilisateur contient désormais un exemple basé sur des cas d'utilisation qui montre comment héberger la sortie de build dans un compartiment S3. L'exemple s'appuie sur la prise en charge récente des artefacts de génération non chiffrés. Pour de plus amples informations, veuillez consulter [Création d'un site web statique avec la sortie de génération hébergée dans un compartiment Amazon S3](#).

14 août 2018

[Support pour remplacer le nom d'un artefact par le versionnement sémantique](#)

Vous pouvez désormais utiliser le versionnement sémantique pour spécifier un format CodeBuild utilisé pour nommer les artefacts de construction. Cela peut s'avérer utile, car un artefact de génération avec un nom codé de manière fixe remplacera les artefacts de génération précédents qui utilisent le même nom codé de manière fixe. Par exemple, si une génération est déclenchée plusieurs fois par jour, vous pouvez désormais ajouter un horodatage à son nom d'artefact. Chaque nom d'artefact de génération est unique, ce qui évite d'écraser les artefacts des générations précédentes.

7 août 2018

[Support des artefacts de construction non chiffrés](#)

CodeBuild prend désormais en charge les builds avec des artefacts de build non chiffrés. Pour plus d'informations, voir [Création d'un projet de génération \(console\)](#).

26 juillet 2018

[Support pour les CloudWatch métriques et les alarmes Amazon](#)

CodeBuild fournit désormais une intégration avec CloudWatch les métriques et les alarmes. Vous pouvez utiliser la CloudWatch console CodeBuild or pour surveiller les builds au niveau du projet et du compte. Pour de plus amples informations, veuillez consulter [Surveillance des générations](#).

19 juillet 2018

[Support pour signaler l'état d'une construction](#)

CodeBuild peut désormais signaler l'état du début et de la fin d'une compilation à votre fournisseur source. Pour de plus amples informations, veuillez consulter [Création d'un projet de génération dans CodeBuild](#).

10 juillet 2018

[Variables d'environnement ajoutées à CodeBuild la documentation](#)

La page [Variables d'environnement dans les environnements de génération](#) a été mise à jour avec les variables d'environnement CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH et CODEBUILD_START_TIME.

9 juillet 2018

[Support pour un finally bloc dans le fichier buildspec](#)

La CodeBuild documentation a été mise à jour avec des détails sur le finally bloc facultatif dans un fichier buildspec. Les commandes du bloc final sont toujours exécutées après les commandes du bloc de commandes correspondant. Pour de plus amples informations, veuillez consulter [Syntaxe des spécifications de génération](#).

le 20 juin 2018

[CodeBuild notifications de mise à jour des agents](#)

La CodeBuild documentation a été mise à jour avec des informations détaillées sur la manière dont vous pouvez utiliser Amazon SNS pour être averti lorsque de nouvelles versions de l'agent CodeBuild sont publiées. Pour plus d'informations, voir [Recevoir des notifications pour les nouvelles versions de AWS CodeBuild l'agent](#).

15 juin 2018

Mises à jour antérieures

Le tableau ci-après décrit des modifications importantes apportées dans chaque version du Guide de l'utilisateur AWS CodeBuild avant juin 2018.

| Modification | Description | Date |
|------------------------------------|---|-------------|
| Prise en charge des builds Windows | CodeBuild prend désormais en charge les versions pour | 25 mai 2018 |

| Modification | Description | Date |
|--|---|-------------|
| | la plate-forme Microsoft Windows Server, y compris un environnement de génération préconfiguré pour .NET Core 2.0 sous Windows. Pour plus d'informations, consultez Exemples Microsoft Windows pour CodeBuild . | |
| Prise en charge de l'idempotence des générations | Lorsque vous exécutez la commande <code>start-build</code> avec l'AWS Command Line Interface (AWS CLI), vous pouvez indiquer que la génération est idempotente. Pour plus d'informations, consultez Exécution d'une génération (AWS CLI) . | 15 mai 2018 |
| Prise en charge du remplacement d'un plus grand nombre de paramètres d'un projet de génération | Vous pouvez désormais remplacer plusieurs paramètres d'un projet de génération lorsque vous créez une génération. Les remplacements concernent alors uniquement cette génération. Pour plus d'informations, consultez Exécution d'une génération dans AWS CodeBuild . | 15 mai 2018 |

| Modification | Description | Date |
|--|---|--------------|
| Prise en charge des points de terminaison VPC | Vous pouvez désormais utiliser les points de terminaison d'un VPC pour améliorer la sécurité de vos générations. Pour plus d'informations, consultez Utilisation de points de terminaison de VPC . | 18 mars 2018 |
| Prise en charge des déclencheurs | Vous pouvez désormais créer des déclencheurs afin de programmer des générations à intervalles réguliers. Pour plus d'informations, consultez Création de déclencheurs AWS CodeBuild . | 28 mars 2018 |
| Documentation sur les points de terminaison FIPS | Vous pouvez maintenant apprendre à utiliser le AWS Command Line Interface (AWS CLI) ou un AWS SDK pour savoir comment utiliser l'un des quatre points CodeBuild de terminaison FIPS (Federal Information Processing Standards). Pour plus d'informations, consultez Spécification du point de terminaison AWS CodeBuild . | 28 mars 2018 |

| Modification | Description | Date |
|--|--|------------------|
| AWS CodeBuild disponible en Asie-Pacifique (Mumbai), en Europe (Paris) et en Amérique du Sud (São Paulo) | AWS CodeBuild est désormais disponible dans les régions Asie-Pacifique (Mumbai), Europe (Paris) et Amérique du Sud (São Paulo). Pour de plus amples informations, veuillez consulter AWS CodeBuild dans le Référence générale d'Amazon Web Services. | 28 mars 2018 |
| GitHub Support pour serveurs d'entreprise | CodeBuild peut désormais créer à partir du code source stocké dans un référentiel GitHub Enterprise Server. Pour plus d'informations, consultez GitHub Exemple de serveur d'entreprise . | 25 janvier 2018 |
| Prise en charge de la profondeur du clone Git | CodeBuild prend désormais en charge la création d'un clone superficiel avec un historique tronqué au nombre de validations spécifié. Pour plus d'informations, consultez Création d'un projet de génération . | 25 janvier 2018 |
| Prise en charge de VPC | Les générations activées pour VPC sont désormais en mesure d'accéder aux ressources à l'intérieur de votre VPC. Pour plus d'informations, consultez Prise en charge de VPC . | 27 novembre 2017 |

| Modification | Description | Date |
|---|---|------------------|
| Prise en charge de la mise en cache des dépendances | CodeBuild prend désormais en charge la mise en cache des dépendances. Cela permet d' CodeBuild enregistrer certaines parties réutilisables de l'environnement de construction dans le cache et de les utiliser entre les versions. | 27 novembre 2017 |
| Prise en charge des badges de génération | CodeBuild prend désormais en charge l'utilisation de badges de construction, qui fournissent une image intégrée et générée dynamiquement (badge) qui affiche l'état de la dernière version d'un projet. Pour plus d'informations, consultez Exemple de badges de génération . | 27 novembre 2017 |
| AWS Config intégration | AWS Config prend désormais en charge en CodeBuild tant que AWS ressource, ce qui signifie que le service peut suivre vos CodeBuild projets. Pour plus d'informations sur AWS Config, voir AWS Config échantillon . | 20 octobre 2017 |

| Modification | Description | Date |
|---|--|-------------------|
| Reconstruire automatiquement le code source mis à jour dans les GitHub référentiels | Si votre code source est stocké dans un GitHub référentiel, vous pouvez activer la reconstruction AWS CodeBuild de votre code source chaque fois qu'une modification de code est envoyée au référentiel. Pour plus d'informations, consultez GitHub exemple de pull request et de filtre Webhook . | 21 septembre 2017 |

| Modification | Description | Date |
|--|--|-------------------|
| Nouvelles méthodes de stockage et de récupération de variables d'environnement sensibles ou volumineuses dans Amazon EC2 Systems Manager Parameter Store | <p>Vous pouvez désormais utiliser la AWS CodeBuild console ou le AWS CLI pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans Amazon EC2 Systems Manager Parameter Store. Vous pouvez également utiliser la AWS CodeBuild console pour stocker ces types de variables d'environnement dans Amazon EC2 Systems Manager Parameter Store. Auparavant, vous pouviez uniquement récupérer ces types de variables d'environnement en les incluant dans une spécification de génération ou en exécutant des commandes de génération pour automatiser l' AWS CLI. Vous ne pouviez stocker ces types de variables d'environnement qu'à l'aide de la console Amazon EC2 Systems Manager Parameter Store. Pour plus d'informations, reportez-vous aux Modification des paramètres d'un projet de génération sections Création d'un projet de génération, et Exécution d'une génération.</p> | 14 septembre 2017 |

| Modification | Description | Date |
|--|--|-----------------|
| Prise en charge de la suppression de génération | Vous pouvez désormais supprimer des générations dans AWS CodeBuild. Pour plus d'informations, consultez Suppression de générations . | 31 août 2017 |
| Méthode mise à jour pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans le magasin de paramètres Amazon EC2 Systems Manager à l'aide d'un buildspec | AWS CodeBuild facilite désormais l'utilisation d'une spécification de construction pour récupérer des variables d'environnement sensibles ou volumineuses stockées dans le magasin de paramètres Amazon EC2 Systems Manager. Auparavant, vous pouviez uniquement récupérer ces types de variables d'environnement en exécutant des commandes de génération pour automatiser l' AWS CLI. Pour plus d'informations, consultez le parameter-store mappage dans Syntaxe d'un fichier buildspec . | le 10 août 2017 |
| AWS CodeBuild prend en charge Bitbucket | CodeBuild peut désormais être compilé à partir du code source stocké dans un dépôt Bitbucket. Pour plus d'informations, reportez-vous Création d'un projet de génération aux sections et Exécution d'une génération . | le 10 août 2017 |

| Modification | Description | Date |
|--|---|--------------|
| AWS CodeBuild disponible dans l'ouest des États-Unis (Californie du Nord), en Europe (Londres) et au Canada (centre) | AWS CodeBuild est désormais disponible dans les régions de l'ouest des États-Unis (Californie du Nord), de l'Europe (Londres) et du Canada (centre). Pour de plus amples informations, veuillez consulter AWS CodeBuild dans le Référence générale d'Amazon Web Services. | 29 juin 2017 |
| Autres noms et emplacements de fichiers de spécification de génération pris en charge | Vous pouvez désormais spécifier un autre nom de fichier ou emplacement pour un fichier de spécification de génération à utiliser pour un projet de génération au lieu d'un fichier de spécification de génération par défaut nommé <code>buildspec.yml</code> à la racine du code source. Pour plus d'informations, consultez Nom de fichier buildspec et emplacement de stockage . | 27 juin 2017 |

| Modification | Description | Date |
|--|---|--------------|
| Exemple de notifications de génération mis à jour | CodeBuild fournit désormais un support intégré pour les notifications de build via Amazon CloudWatch Events et Amazon Simple Notification Service (Amazon SNS). L'exemple Exemple de notifications de génération précédent a été mis à jour pour montrer ce nouveau comportement. | 22 juin 2017 |
| Exemple Docker dans une image personnalisée ajouté | Un exemple montrant comment utiliser CodeBuild une image Docker personnalisée pour créer et exécuter une image Docker a été ajouté. Pour plus d'informations, consultez le Exemple Docker dans une image personnalisée . | 7 juin 2017 |

| Modification | Description | Date |
|---|---|-------------|
| Récupère le code source pour les GitHub pull requests | Lorsque vous exécutez un build CodeBuild qui repose sur le code source stocké dans un GitHub référentiel, vous pouvez désormais spécifier un ID de GitHub pull request à compiler. Vous pouvez également spécifier plutôt un ID de validation, un nom de branche ou un nom de balise. Pour plus d'informations, voir la valeur de la version source dans Exécution d'une génération (console) ou la <code>sourceVersion</code> valeur dans Exécution d'une génération (AWS CLI) . | 6 juin 2017 |

| Modification | Description | Date |
|---|--|------------|
| Version de spécification de génération mise à jour | Une nouvelle version du format de spécification de génération a été publiée. La version 0.2 résout le problème de l' CodeBuild exécution de chaque commande de construction dans une instance distincte du shell par défaut. En outre, dans la version 0.2, <code>environment_variables</code> est renommé en <code>env</code> et <code>plaintext</code> est renommé en <code>variables</code> . Pour plus d'informations, consultez Référence de spécification de construction pour CodeBuild . | 9 mai 2017 |
| Dockerfiles pour créer des images disponibles dans GitHub | Les définitions de nombreuses images de build fournies par AWS CodeBuild sont disponibles sous forme de Dockerfiles dans GitHub. Pour plus d'informations, consultez la colonne Définition du tableau dans Images Docker fournies par CodeBuild . | 2 mai 2017 |

| Modification | Description | Date |
|--|--|----------------|
| AWS CodeBuild disponible en Europe (Francfort), en Asie-Pacifique (Singapour), en Asie-Pacifique (Sydney) et en Asie-Pacifique (Tokyo) | AWS CodeBuild est désormais disponible dans les régions Europe (Francfort), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney) et Asie-Pacifique (Tokyo). Pour de plus amples informations, veuillez consulter AWS CodeBuild dans le Référence générale d'Amazon Web Services. | 21 mars 2017 |
| CodePipeline support aux actions de test pour CodeBuild | Vous pouvez désormais ajouter à un pipeline dans CodePipeline une action de test qui utilise CodeBuild. Pour plus d'informations, consultez Ajout d'une action de test CodeBuild à un pipeline (console CodePipeline) . | 8 mars 2017 |
| Les fichiers de spécification de génération prennent en charge l'extraction de la sortie de génération depuis des répertoires de niveau supérieur sélectionnés | Les fichiers Buildspec vous permettent désormais de spécifier des répertoires individuels de premier niveau dont vous pouvez demander l'inclusion dans les artefacts de sortie CodeBuild de construction. Pour ce faire, vous devez utiliser le mappage <code>base-directory</code> . Pour plus d'informations, consultez Syntaxe d'un fichier buildspec . | 8 février 2017 |

| Modification | Description | Date |
|---|--|-----------------|
| Variables d'environnement intégrées | AWS CodeBuild fournit des variables d'environnement intégrées supplémentaires que vos builds peuvent utiliser. Il s'agit notamment de variables d'environnement qui décrivent l'entité qui a démarré la génération, l'URL vers le référentiel du code source, l'ID de version du code source, etc. Pour plus d'informations, consultez Variables d'environnement dans les environnements de génération . | 30 janvier 2017 |
| AWS CodeBuild disponible dans l'est des États-Unis (Ohio) | AWS CodeBuild est désormais disponible dans la région USA Est (Ohio). Pour de plus amples informations, veuillez consulter AWS CodeBuild dans le Référence générale d'Amazon Web Services. | 19 janvier 2017 |

| Modification | Description | Date |
|--|---|-----------------|
| Informations sur les comportements de shell et de commande | CodeBuild exécute chaque commande que vous spécifiez dans une instance distincte du shell par défaut d'un environnement de génération. Ce comportement par défaut peut produire des effets inattendus sur vos commandes. Nous vous recommandons certaines approches pour contourner ce comportement par défaut si nécessaire. Pour plus d'informations, consultez Shells et commandes dans les environnements de génération . | 9 décembre 2016 |
| Informations sur les variables d'environnement | CodeBuild fournit plusieurs variables d'environnement que vous pouvez utiliser dans vos commandes de compilation. Vous pouvez également définir vos propres variables d'environnement. Pour plus d'informations, consultez Variables d'environnement dans les environnements de génération . | 7 décembre 2016 |
| Rubrique de dépannage | Des informations de dépannage sont désormais disponibles. Pour plus d'informations, consultez Résolution des problèmes AWS CodeBuild . | 5 décembre 2016 |

| Modification | Description | Date |
|--|--|-------------------|
| Version initiale du plug-in Jenkins | Il s'agit de la version initiale du plugin CodeBuild Jenkins. Pour plus d'informations, consultez Utilisation d'AWS CodeBuild avec Jenkins . | 5 décembre 2016 |
| Version initiale du Guide de l'utilisateur | Il s'agit de la première édition du guide de l'utilisateur CodeBuild . | 1er décembre 2016 |

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.