



Guide de l'utilisateur

AWS CloudFormation



Version de l'API 2010-05-15

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

AWS CloudFormation: Guide de l'utilisateur

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

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

Table of Contents

Qu'est-ce que c'est CloudFormation ?	1
Simplification de la gestion de l'infrastructure	1
Réplication rapide de votre infrastructure	1
Contrôle et suivi faciles des modifications apportées à votre infrastructure	2
Commencer avec CloudFormation	2
Informations connexes	2
Prise en main	4
Comment CloudFormation fonctionne	4
Concepts clés	4
Comment CloudFormation fonctionne	8
Façons de démarrer avec CloudFormation	12
S'inscrire à un Compte AWS	12
Inscrivez-vous pour un Compte AWS	12
Création d'un utilisateur doté d'un accès administratif	13
Création de votre première pile	14
Conditions préalables	15
Création d'une CloudFormation pile avec la console	15
Surveiller la création de la pile	19
Tester le service Web	20
Résolution des problèmes	21
Nettoyage	22
Étapes suivantes	23
Bonnes pratiques	24
Réduction du cycle de rétroaction pour accélérer le développement	25
Organisation des piles par cycle de vie et propriétaire	26
Utilisation des références inter-piles pour récupérer la valeur d'une sortie exportée par une autre pile	27
Utilisation CloudFormation StackSets pour les déploiements multicomptes et multirégions	27
Vérification des quotas pour tous les types de ressource	28
Réutilisation de modèles pour répliquer des piles dans plusieurs environnements	28
Utiliser des modules pour réutiliser les configurations de ressources	29
Adoption des pratiques d'infrastructure en tant que code	29
N'intégrez pas d'informations d'identification dans vos modèles	30
Utiliser des AWS types de paramètres spécifiques	30

Utilisation de contraintes de paramètre	31
Utiliser des pseudoparamètres pour promouvoir la portabilité	31
Utilisation d'AWS::CloudFormation::Init pour déployer des applications dans des instances Amazon Amazon EC2	32
Utilisation des derniers scripts d'assistant	32
Validation des modèles avant leur utilisation	32
Valider les modèles pour la conformité des politiques de l'organisation	33
Utilisation de YAML ou JSON pour la création de modèles	33
Mise en œuvre d'une stratégie de balisage complète	34
Exploitation des macros de modèle pour des transformations avancées	34
Gérez toutes les ressources de la pile via CloudFormation	35
Création de jeux de modification avant la mise à jour des piles	35
Utilisation des politiques de pile pour protéger les ressources	36
AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels	36
Utilisation des révisions de code et des contrôles de révision pour gérer les modèles	36
Mettez à jour régulièrement vos instances Amazon EC2	37
Utilisation régulière de la détection de dérive	37
Configuration des déclencheurs de restauration automatique	37
Mise en œuvre de stratégies efficaces de refactorisation des piles	37
Utilisez les CloudFormation Hooks pour la gestion du cycle de vie	38
Utilisation du générateur IaC pour créer des modèles à partir de ressources existantes	38
Utilisation AWS Infrastructure Composer pour la conception de modèles visuels	39
Envisagez de l'utiliser AWS Cloud Development Kit (AWS CDK) pour une infrastructure complexe	40
Utilisation du contrôle d'accès IAM	40
Application du principe du moindre privilège	41
Sécurisation des paramètres sensibles	41
Implémentez la politique sous forme de code avec AWS CloudFormation Guard	41
Utilisation des modèles	43
Où sont stockés les modèles	44
Validation des modèles	44
Premiers pas avec les modèles	2
Prévoyez d'utiliser la référence du CloudFormation modèle	45
Exemples de modèle	46
Format du modèle	46
Structure du modèle	47

Commentaires	49
Spécifications	50
En savoir plus	50
Utilisation des expressions régulières	51
Sections de modèle	52
Resources	53
Parameters	62
Outputs	76
Mappings	82
Metadata	90
Rules	96
Conditions	116
Transform	129
Version de format	130
Description	130
Compositeur d'infrastructure	131
Pourquoi utiliser Infrastructure Composer en mode CloudFormation console ?	131
En quoi ce mode est-il différent de celui de la console Infrastructure Composer ?	132
AWS CloudFormation Serveur de langues	132
IDEs intégration avec le serveur de AWS CloudFormation langue	133
Prise en main	133
Initialisation d'un CloudFormation projet dans l'IDE (VS Code uniquement)	140
Open source	141
Vous avez besoin d'aide ?	142
Générateur IaC	142
Considérations	143
Permissions	144
Commandes couramment utilisées	144
Migrer un modèle vers AWS CDK	145
Lancer une analyse des ressources	145
Consulter le résumé du scan	150
Création d'un modèle à partir de ressources numérisées	151
Création d'une pile à partir de ressources numérisées	156
Résolution de propriétés en écriture seule	158
Obtenir des valeurs stockées dans d'autres services	163
Considérations d'ordre général	164

Obtenir la valeur en texte brut de Systems Manager	165
Obtenir la chaîne sécurisée de Systems Manager	169
Profitez de Secrets Manager	172
Obtenir des AWS valeurs	177
Syntaxe	177
Pseudo-paramètres disponibles	178
Exemples	180
Obtenir les sorties de la pile	184
Exportation de valeurs de sortie de pile comparée à l'utilisation de piles imbriquées	185
Considérations	185
Affichage des valeurs de sortie exportées	186
Affichage des piles qui importent une valeur de sortie exportée	187
Spécification de ressources existantes au moment de l'exécution	188
Présentation de	188
Exemple	189
Considérations	193
Types de paramètres AWS spécifiques pris en charge	194
Types de paramètres Systems Manager pris en charge	196
Types de paramètres Systems Manager non pris en charge	196
Procédures	197
Référez-vous aux sorties de ressources dans une autre CloudFormation pile	198
Déploiement d'applications sur Amazon EC2	203
Mettre à jour une pile	213
Création d'une application redimensionnée et équilibrée de charge	230
Appairage avec un VPC dans un autre compte	244
Réaliser des blue/green déploiements ECS	255
Extraits de modèle	279
Général	280
Auto scaling (Mise à l'échelle automatique)	291
AWS Console de facturation	324
CloudFormation	327
CloudFront	334
CloudWatch	347
CloudWatch Journaux	353
DynamoDB	387
Amazon EC2	391

Amazon ECS	429
Amazon EFS	452
Elastic Beanstalk	477
Elastic Load Balancing	482
IAM	487
AWS Lambda	505
Amazon Redshift	512
Amazon RDS	522
Route 53	531
Amazon S3	539
Amazon SNS	548
Amazon SQS	549
Amazon Timestream	549
Piles basées sur Windows	578
Amorçage des piles Windows	580
Utiliser les CloudFormation types de ressources fournis	587
Ressources personnalisées	589
Macros de modèle	635
Piles imbriquées	653
Conditions d'attente	660
Création de configurations de ressources réutilisables avec des modules	666
Considérations relatives à l'utilisation de modules	668
Compréhension de la gestion des versions des modules	668
Utiliser des modules dans un modèle	670
Utilisez des paramètres pour spécifier les valeurs des modules	671
Ressources du module de référence	674
Création et gestion de piles	676
Interfaces pour gérer vos piles	678
Création d'une pile	679
Création d'une pile	679
Configuration des options de pile	682
Prévisualisation de la configuration de la pile	685
Affichage des informations d'une pile	686
Mettre à jour votre modèle de pile	689
Comprendre les comportements de mise à jour des ressources d'une pile	691
Mettre à jour les piles à l'aide de jeux de modifications	693

Créer un jeu de modifications	695
Afficher un jeu de modifications	702
Ensembles de modifications sensibles à la dérive	710
Exécuter un jeu de modifications	717
Supprimer un jeu de modifications	719
Exemples de jeux de modifications	721
Modifier les ensembles pour les piles imbriquées	734
Valider les déploiements de stack	741
Comment fonctionne la validation préalable au déploiement	741
Considérations	742
Conditions préalables	742
Valider un déploiement de stack (console)	743
Valider un déploiement de stack (AWS CLI)	743
Types de validation	745
Limite de ressources	746
Mettre à jour directement les piles	760
Annuler la mise à jour d'une pile	764
Pour annuler une mise à jour de pile (console)	764
Pour annuler une mise à jour de pile (AWS CLI)	765
Supprimer une pile	765
Ressources connexes	767
Affichage des piles supprimées	767
Surveiller la progression de la pile	768
Afficher les événements de la pile	768
Afficher les événements de la pile par opération	777
Afficher le graphique de la chronologie du déploiement de la pile	781
Comprendre les événements de création de pile	784
Surveiller les mises à jour de pile	786
Poursuite de la restauration d'une mise à jour	789
Détermination de la cause d'un échec de pile	793
Options d'échec de pile	794
Restaurer votre pile en cas de dépassement d'alarme	804
Ajouter des déclencheurs de restauration lors de la création ou de la mise à jour d'une pile	805
Ajouter des déclencheurs de restauration à un jeu de modifications	806
Afficher les déclencheurs de restauration pour une pile	807

Afficher les déclencheurs de restauration pour un jeu de modifications	807
Détectez les modifications de configuration non gérées apportées aux piles et aux ressources	807
Qu'est-ce qu'un écart ?	808
Codes de statut de détection de l'écart	809
Considérations lors de la détection de l'écart	812
Détectez la dérive sur l'ensemble d'une CloudFormation pile	815
Détecter l'écart sur des ressources de pile individuelles	820
Résoudre l'écart avec une opération d'importation	824
Importer AWS des ressources	835
Importer manuellement des ressources AWS	836
Importer automatiquement des ressources AWS	875
Rétablissement d'une opération d'importation	877
Refactorisation de piles	882
Comment fonctionne la refactorisation de pile	883
Considérations sur la refactorisation des piles	884
Conditions préalables	884
Refactor Stacks (console)	885
Piles de refactorisation (AWS CLI)	886
Limite de ressources	891
Prise en charge des types de ressources	895
Utilisez des liens de création rapide pour créer des piles CloudFormation	1090
Format d'URL	1091
Exemple	1092
Création d'une pile à l'aide d'un lien de création rapide	1093
Exemples de commandes pour AWS CLI et PowerShell	1094
CancelUpdateStack	1095
ContinueUpdateRollback	1096
CreateStack	1097
Déploiement	1101
DeleteStack	1102
DescribeStackEvents	1103
DescribeStackResource	1105
DescribeStackResources	1107
DescribeStacks	1110
GetTemplate	1114

ListStackResources	1115
ListStacks	1117
UpdateStack	1119
ValidateTemplate	1123
Charger des artefacts locaux dans un compartiment S3	1126
Gérer les piles avec StackSets	1130
StackSets concepts	1131
Comptes d'administrateur et comptes de destination	1132
CloudFormation StackSets	1132
Modèles d'autorisation pour StackSets	1132
Instances de piles	1133
StackSet opérations	1134
StackSet options de fonctionnement	1135
Étiquettes	1137
StackSets codes de statut	1138
Codes d'état de l'instance de pile	1139
Conditions préalables	1140
Activez Régions AWS ceux qui sont désactivés par défaut	1141
Accorder des autorisations autogérées	1142
Activer l'accès approuvé	1159
Commencez à utiliser un modèle type	1164
Conditions préalables	1165
Créez un StackSet avec un exemple de modèle à partir de la console	1165
StackSet Création de moniteurs	1168
Afficher les StackSet résultats	1169
Mettez à jour votre StackSet	1169
Ajoutez des piles à votre StackSet	1170
Nettoyage	1171
Étapes suivantes	1173
Créer StackSets (autorisations autogérées)	1174
Créer un StackSet avec des autorisations autogérées (console)	1174
Créer un StackSet avec des autorisations autogérées ()AWS CLI	1176
Créer StackSets (autorisations gérées par le service)	1178
Considérations	1178
Créer un StackSet avec des autorisations gérées par le service (console)	1179
Créer un StackSet avec des autorisations gérées par le service ()AWS CLI	1182

Activer/désactiver les déploiements automatiques	1186
Fonctionnement des déploiements automatiques	1186
Considérations	1187
Activer ou désactiver les déploiements automatiques (console)	1187
Activer ou désactiver les déploiements automatiques (AWS CLI)	1188
Mettre à jour StackSets	1189
Mettez à jour votre StackSet (console)	1189
Mettez à jour votre StackSet (AWS CLI)	1191
Ajoutez des piles à StackSets	1193
Ajouter des piles à une StackSet (console)	1193
Ajouter des piles à un StackSet (AWS CLI)	1195
Remplacer les paramètres sur les piles	1196
Remplacer les paramètres sur les piles (console)	1197
Remplacer les paramètres sur les piles (AWS CLI)	1199
Supprimer des piles de StackSets	1200
Supprimer des piles de votre StackSet (console)	1201
Supprimer les piles de votre StackSet (AWS CLI)	1203
Supprimer StackSets	1204
Supprimer une StackSet (console)	1205
Supprimer un StackSet (AWS CLI)	1205
Supprimer les rôles de service (facultatif)	1206
Portails de compte de destination	1207
Exigences	1207
CloudFormation modèles pour créer des fonctions Lambda	1208
Choix du mode simultané	1208
Fonctionnement de chaque mode simultané	1209
Sélection de la tolérance stricte aux pannes et de la tolérance souple aux pannes en fonction de la vitesse de déploiement	1212
Choix du mode simultané (console)	1214
Choix du mode simultané (AWS CLI)	1214
Détection de la dérive StackSets	1215
Comment CloudFormation effectue la détection de la dérive sur un StackSet	1216
Détecter la dérive sur une StackSet (console)	1217
Détecter la dérive sur un StackSet (AWS CLI)	1218
Arrêter la détection de dérive sur un StackSet	1225
Importez des piles dans StackSets	1226

Importation de piles autogérées	1227
Importation de pile gérée par le service	1230
Revenir sur les importations de piles	1236
Bonnes pratiques	1236
Définition du modèle	1237
Création ou ajout de piles dans StackSet	1237
Mettre à jour les piles dans un StackSet	1237
Exemples de modèle	1238
Résolution des problèmes	1240
Raisons les plus courantes de l'échec de l'opération de pile	1240
Nouvelle tentative non réussie de création ou mise à jour de pile	1241
Echec de la suppression de l'instance de pile	1242
L'opération d'importation de pile échoue	1242
Nombre de défaillances des instances de Stack pour les StackSets opérations	1242
Synchronisation des piles avec le code source Git	1249
Comment fonctionne la synchronisation Git	1250
Comment fonctionne la synchronisation Git	1250
Commentaires sur les demandes d'extraction	1252
Fichier de déploiement de pile	1252
CloudFormation fichier modèle	1253
Référentiel de définition du modèle	1253
Conditions préalables	1254
Référentiel Git	1254
CloudFormation modèle	1254
Rôle de service de synchronisation Git	1254
Autorisations IAM pour les utilisateurs de la console	1258
Créer une pile à partir du code source du référentiel	1261
Créer une pile à partir du code source du référentiel	1261
Mettez à jour votre pile depuis votre référentiel Git	1264
Activation des commentaires sur les requêtes pull	1264
Tableau de bord d'état	1265
État de la synchronisation Git	1265
Derniers événements de synchronisation	1266
États de pile pris en charge	1267
Gestion des extensions avec le CloudFormation registre	1271
Documents associés	1271

Concepts	1272
Types d'extension	1272
Extensions publiques	1273
Extensions activées	1273
Afficher les extensions disponibles et activées	1274
Extensions publiques	1275
Rôle IAM	1276
Utiliser automatiquement les nouvelles versions des extensions	1279
Utilisez des alias pour faire référence aux extensions	1279
AWS CLI Commandes couramment utilisées pour travailler avec des extensions publiques	1280
Activer une extension publique	1280
Mettre à jour une extension publique	1285
Désactivation des extensions publiques	1286
Extensions privées	1288
Autorisations IAM	1288
AWS CLI Commandes couramment utilisées pour travailler avec des extensions privées ..	1288
Enregistrer une extension privée	1289
Désenregistrer les extensions privées	1291
Modifier les données de configuration pour les extensions	1293
Autorisations requises pour utiliser des références dynamiques	1294
Modifier les données de configuration d'une extension (console)	1294
Modifier les données de configuration d'une extension (AWS CLI)	1295
Enregistrer les types de ressources dans AWS Config	1296
Prévention de l'enregistrement de propriétés sensibles dans un élément de configuration .	1297
Livraison continue	1298
Procédure : Création d'un pipeline pour les piles de test et de production	1298
Conditions préalables	1299
Présentation de la procédure	1299
Étape 1 : Modifier l'artefact et le charger dans un compartiment S3	1300
Étape 2 : Créer la pile du pipeline	1302
Étape 3 : Afficher la WordPress pile	1308
Étape 4 : Nettoyer les ressources	1308
Consultez aussi	1309
Référence sur les propriétés de configuration	1310
Propriétés de configuration (Console)	1310

Propriétés de configuration (objet JSON)	1313
Consultez aussi	1318
CloudFormation artefacts	1319
Fichier de modèle de pile	1319
Fichier de configuration de modèle	1319
Consultez aussi	1321
Utilisation de fonctions de remplacement de paramètres avec des pipelines CodePipeline	1321
Fn::GetArtifactAtt	1321
Fn::GetParam	1323
Consultez aussi	1326
Guide de référence des modèles	1327
Sécurité	1328
Protéger les piles contre la suppression	1329
Contrôle des personnes autorisées à modifier la protection contre la résiliation sur les piles	1331
Empêcher les mises à jour des ressources de la pile	1331
Exemple de politique de pile	1332
Définition d'une politique de pile	1333
Définition d'une politique de pile	1338
Mise à jour des ressources protégées	1339
Modification d'une politique de pile	1342
Autres exemples de politiques de pile	1343
Protection des données	1347
Chiffrement au repos	1348
Chiffrement en transit	1348
Confidentialité du trafic inter-réseau	1348
Contrôle de l'accès avec IAM	1349
Définition de politiques basées sur l'identité IAM pour CloudFormation	1349
Reconnaissance des ressources IAM dans les modèles CloudFormation	1359
Gestion des informations d'identification pour les applications exécutées dans des instances Amazon EC2	1360
Autorisation d'un accès temporaire (accès fédéré)	1361
Exemples de stratégies basées sur l'identité	1362
AWS politiques gérées	1370
CloudFormation rôle de service	1373
Prévention du problème de l'adjoind confus entre services	1374

Demandes FAS et évaluation des autorisations	1377
Journalisation des appels d'API	1379
CloudFormation informations dans CloudTrail	1380
Comprendre les entrées du fichier CloudFormation journal	1381
Sécurité de l'infrastructure	1385
Résilience	1385
Validation de conformité	1386
Analyse de la configuration et des vulnérabilités	1386
Bonnes pratiques de sécurité	1386
Utilisation du contrôle d'accès IAM	1387
N'incorporez pas d'informations d'identification dans vos modèles	1387
AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels	1388
AWS PrivateLink	1388
Considérations relatives aux points de CloudFormation terminaison VPC	1388
Création d'un point de terminaison VPC d'interface pour CloudFormation	1389
Création d'une politique de point de terminaison VPC pour CloudFormation	1390
Surveillance avec EventBridge	1392
CloudFormation et présentation des événements de synchronisation Git	1393
Permissions	1395
Création d'un modèle d'événement personnalisé	1395
Référence détaillée des événements	1396
Changement de statut des ressources	1398
Changement d'état de la pile	1400
Changement d'état de la détection d'écart	1402
StackSet Changement de statut	1405
StackSet Modification du statut de l'instance Stack	1407
StackSetModification du statut de l'opération	1409
Modification de l'état de synchronisation du référentiel	1411
Modification de l'état de synchronisation des ressources	1414
Quotas	1417
Disponibilité des fonctions	1426
StackSets et macros	1426
Résolution des problèmes	1427
Guide de dépannage	1427
Résolution des erreurs	1428
Échec de la suppression de la pile	1429

Erreur de dépendance	1430
AWS Config et AWS Systems Manager conflits	1430
Erreur d'analyse d'un paramètre lors de la transmission d'une liste	1431
Autorisations IAM insuffisantes	1431
Valeur non valide ou propriété de ressource non prise en charge	1431
Quota dépassé	1432
Échec de la restauration des piles imbriquées	1432
Aucune mise à jour à effectuer	1433
La ressource n'a pas pu se stabiliser lors d'une opération de création, de mise à jour ou de suppression de pile	1433
Le groupe de sécurité n'existe pas dans le VPC	1434
Échec de la restauration de la mise à jour	1434
La condition d'attente n'a pas reçu le nombre de signaux requis à partir d'une instance Amazon EC2	1436
Ressource retirée de la pile mais non supprimée	1437
Contacter l'assistance	1437
Résolution des problèmes avec Amazon Q	1439
Caractéristiques	1439
Comment ça marche	1440
Tarification	1440
Disponibilité par région	1440
Limitations	1440
Historique de la documentation	1441
Mises à jour archivées	1471
.....	mdxcv

Qu'est-ce que c'est CloudFormation ?

CloudFormation est un service qui vous aide à modéliser et à configurer vos AWS ressources afin que vous puissiez passer moins de temps à gérer ces ressources et plus de temps à vous concentrer sur les applications qui s'exécutent AWS. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez (comme les instances Amazon EC2 ou les instances de base de données Amazon RDS), et vous vous CloudFormation occupez du provisionnement et de la configuration de ces ressources pour vous. Vous n'avez pas besoin de créer et de configurer AWS des ressources individuellement et de déterminer ce qui dépend de quoi ; il CloudFormation gère tout cela. Les scénarios suivants montrent comment CloudFormation vous pouvez vous aider.

Simplification de la gestion de l'infrastructure

Pour une application web évolutive qui inclut également une base de données dorsale, vous pouvez utiliser un groupe Auto Scaling, un équilibreur de charge Elastic Load Balancing et une instance de base de données Amazon Relational Database Service. Vous pouvez utiliser chaque service pour configurer ces ressources et une fois que vous créez les ressources, vous devez les configurer afin qu'elles interagissent. Toutes ces tâches peuvent générer des difficultés et vous faire perdre beaucoup de temps avant même que votre application ne soit opérationnelle.

Au lieu de cela, vous pouvez créer un CloudFormation modèle ou modifier un modèle existant. Un modèle décrit l'ensemble de vos ressources et leurs propriétés. Lorsque vous utilisez ce modèle pour créer une CloudFormation pile, CloudFormation provisionnez le groupe Auto Scaling, l'équilibreur de charge et la base de données pour vous. Une fois la pile créée avec succès, vos AWS ressources sont opérationnelles. Vous pouvez supprimer la pile tout aussi facilement, ce qui supprime toutes les ressources qu'elle contient. En utilisant CloudFormation, vous gérez facilement un ensemble de ressources en tant qu'unité unique.

Réplication rapide de votre infrastructure

Si votre application nécessite une plus grande disponibilité, vous pouvez la répliquer dans plusieurs régions. Dès lors, si une région devient indisponible, vos utilisateurs peuvent continuer à utiliser votre application dans d'autres régions. La difficulté que présente la réplication de votre application est que vous devez également répliquer les ressources. Non seulement vous devez enregistrer toutes les ressources dont votre application a besoin, mais vous devez également mettre en service et configurer ces ressources dans chaque région.

Réutilisez votre CloudFormation modèle pour créer vos ressources de manière cohérente et reproductible. Pour réutiliser votre modèle, il vous suffit de décrire les ressources une seule fois, puis de les mettre en service autant de fois que vous le souhaitez dans plusieurs régions.

Contrôle et suivi faciles des modifications apportées à votre infrastructure

Dans certains cas, vous pouvez avoir besoin de mettre à jour des ressources sous-jacentes façon incrémentielle. Par exemple, vous pouvez passer à un type d'instance plus performant dans votre configuration de lancement Auto Scaling afin de réduire le nombre maximal d'instances dans votre groupe Auto Scaling. Si des problèmes surviennent une fois que vous avez terminé la mise à jour, il est parfois nécessaire de rétablir les paramètres d'origine de votre infrastructure. Pour effectuer cela manuellement, vous devez non seulement vous rappeler quelles ressources ont changé, vous devez également vous souvenir des paramètres d'origine.

Lorsque vous approvisionnez votre infrastructure CloudFormation, le CloudFormation modèle décrit exactement les ressources mises en service et leurs paramètres. Etant donné que ces modèles sont des fichiers de texte, il vous suffit d'analyser les différences entre les modèles pour suivre les modifications apportées à votre infrastructure, tout comme les développeurs contrôlent les révisions apportées au code source. Par exemple, vous pouvez utiliser un système de contrôle de versions avec vos modèles afin de déterminer exactement quelles modifications ont eu lieu, qui les a effectuées et quand. Si, à un moment ou un autre, vous avez besoin d'annuler les modifications apportées à votre infrastructure, vous pouvez utiliser une version précédente de votre modèle.

Commencer avec CloudFormation

CloudFormation est disponible via la CloudFormation [console AWS CLI](#), [AWS SDKs](#), [l'API](#) et via plusieurs intégrations.

Pour une introduction à CloudFormation, voir [Comment CloudFormation fonctionne](#).

Pour commencer à utiliser CloudFormation, voir [Création de votre première pile](#).

Informations connexes

Pour CloudFormation en savoir plus, consultez ce guide de l'utilisateur, ainsi que les ressources suivantes :

- Pour plus de détails sur le produit et FAQs consultez la [page AWS CloudFormation du produit](#).
- Pour en savoir plus sur la tarification, consultez [Tarification AWS CloudFormation](#).

Commencer avec CloudFormation

Vous pouvez commencer à utiliser CloudFormation le en AWS Management Console créant une pile à partir d'un exemple de modèle, qui vous aidera à apprendre les bases de la création de piles. Un modèle est un fichier texte qui définit toutes les ressources d'une pile. Une pile est le déploiement d'un CloudFormation modèle. À partir d'un seul modèle, vous pouvez créer plusieurs piles. Chaque pile contient un ensemble de AWS ressources qui peuvent être gérées comme une seule unité.

CloudFormation est un service gratuit ; toutefois, les AWS ressources que vous incluez dans vos piles vous sont facturées aux taux en vigueur pour chacune d'entre elles. Pour plus d'informations sur AWS les prix, rendez-vous sur la page détaillée de chaque produit sur <http://aws.amazon.com>.

Rubriques

- [Comment CloudFormation fonctionne](#)
- [S'inscrire à un Compte AWS](#)
- [Création de votre première pile](#)

Comment CloudFormation fonctionne

Cette rubrique décrit son CloudFormation fonctionnement et vous présente les concepts clés que vous devez connaître au fur et à mesure de son utilisation.

Rubriques

- [Concepts clés](#)
- [Comment CloudFormation fonctionne](#)
- [Façons de démarrer avec CloudFormation](#)

Concepts clés

Lorsque vous les utilisez CloudFormation, vous travaillez avec des modèles et des piles. Vous créez des modèles pour décrire vos AWS ressources et leurs propriétés. Chaque fois que vous créez une pile, CloudFormation provisionnez les ressources décrites dans votre modèle.

Rubriques

- [Modèles](#)

- [Piles](#)
- [Jeux de modifications](#)

Modèles

Un CloudFormation modèle est un fichier texte au format YAML ou JSON. Vous pouvez enregistrer ces fichiers avec n'importe quelle extension, telle que `.yaml`, `.json`, `.template`, ou `.txt`.

CloudFormation utilise ces modèles comme modèles pour développer vos AWS ressources.

Par exemple, dans un modèle, vous pouvez décrire une instance Amazon EC2, telle que le type d'instance, l'ID d'AMI, les mappages de périphérique de stockage en mode bloc et le nom de sa paire de clés Amazon EC2. Chaque fois que vous créez une pile, vous spécifiez également un modèle que CloudFormation utilise pour créer ce que vous avez décrit dans le modèle.

Par exemple, si vous avez créé une pile avec le modèle suivant, CloudFormation approvisionne une instance avec un ID `ami-0ff8a91507f77f867` AMI, un type d'`t2.micro` instance, un nom de paire de clés `testkey` et un volume Amazon EBS.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A sample template",
```

```

"Resources": {
  "MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "ami-0ff8a91507f77f867",
      "InstanceType": "t2.micro",
      "KeyName": "testkey",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sdm",
          "Ebs": {
            "VolumeType": "io1",
            "Iops": 200,
            "DeleteOnTermination": false,
            "VolumeSize": 20
          }
        }
      ]
    }
  }
}

```

Vous pouvez également spécifier plusieurs ressources dans un modèle unique et les configurer pour interagir les unes avec les autres. Par exemple, vous pouvez modifier le modèle précédent pour inclure une adresse IP Elastic (EIP) et l'associer à l'instance Amazon EC2, comme illustré dans l'exemple suivant :

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1

```

```
        Iops: 200
        DeleteOnTermination: false
        VolumeSize: 20
MyEIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref MyEC2Instance
```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A sample template",
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "t2.micro",
        "KeyName": "testkey",
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/sdm",
            "Ebs": {
              "VolumeType": "io1",
              "Iops": 200,
              "DeleteOnTermination": false,
              "VolumeSize": 20
            }
          }
        ]
      }
    },
    "MyEIP": {
      "Type": "AWS::EC2::EIP",
      "Properties": {
        "InstanceId": {
          "Ref": "MyEC2Instance"
        }
      }
    }
  }
}
```

Les modèles précédents sont centrés sur une seule instance Amazon EC2 ; toutefois, les CloudFormation modèles possèdent des fonctionnalités supplémentaires que vous pouvez utiliser pour créer des ensembles complexes de ressources et réutiliser ces modèles dans plusieurs contextes. Par exemple, vous pouvez ajouter des paramètres d'entrée dont les valeurs sont spécifiées lorsque vous créez une CloudFormation pile. En d'autres termes, vous pouvez définir une valeur telle que le type d'instance lorsque vous créez une pile au lieu de lorsque vous créez le modèle, ce qui facilite la réutilisation du modèle dans différentes situations.

Piles

Lorsque vous utilisez CloudFormation, vous gérez les ressources associées en tant qu'unité unique appelée pile. Pour créer, mettre à jour et supprimer un ensemble de ressources, vous créez, mettez à jour et supprimez des piles. Toutes les ressources d'une pile sont définies par le CloudFormation modèle de la pile. Supposons que vous ayez créé un modèle qui inclut un groupe Auto Scaling, un équilibreur de charge Elastic Load Balancing et une instance de base de données Amazon Relational Database Service (Amazon RDS). Pour créer ces ressources, vous créez une pile en soumettant le modèle que vous avez créé et en CloudFormation provisionnant toutes ces ressources pour vous.

Jeux de modifications

Si vous avez besoin de modifier les ressources en cours d'exécution d'une pile, vous mettez à jour cette dernière. Avant d'apporter des modifications à vos ressources, vous pouvez générer un jeu de modifications, qui représente un résumé de ces modifications. Les jeux de modification vous permettent d'avoir un aperçu de l'impact possible des modifications d'une pile sur les ressources exécutées (y compris les ressources critiques) avant d'appliquer ces modifications.

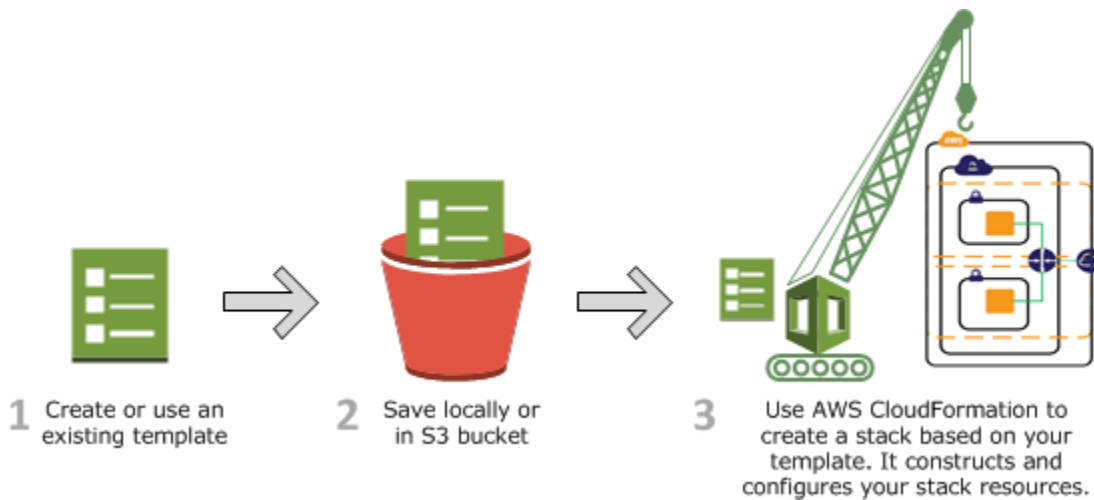
Par exemple, si vous modifiez le nom d'une instance de base de données Amazon RDS, vous CloudFormation créez une nouvelle base de données et supprimerez l'ancienne. Vous perdrez les données de l'ancienne base de données, sauf si vous les avez sauvegardées. Si vous générez un jeu de modifications, vous constaterez que votre modification entraînera le remplacement de votre base de données. Dès lors, vous pourrez prendre les mesures nécessaires avant que de mettre à jour la pile.

Comment CloudFormation fonctionne

Lorsque vous créez CloudFormation votre stack, CloudFormation vous effectuez des appels de service sous-jacents AWS pour provisionner et configurer les ressources décrites dans votre modèle. Vous devez disposer des autorisations nécessaires pour créer ces ressources. Par exemple, pour

créer des instances EC2 à l'aide de CloudFormation, vous devez disposer d'autorisations pour créer des instances. Vous gérez ces autorisations avec [Gestion des identités et des accès AWS](#) (IAM).

Les appels effectués CloudFormation sont tous déclarés par votre modèle. Par exemple, supposons que votre modèle décrit une instance EC2 avec un type d'instance `t2.micro`. Lorsque vous utilisez ce modèle pour créer une pile, CloudFormation appelle l'API de création d'instance Amazon EC2 et spécifie le type d'instance en tant que `t2.micro`. Le schéma suivant résume le flux de travail CloudFormation permettant de créer des piles.



Pour créer une pile

1. Utilisez un éditeur de texte pour créer un CloudFormation modèle au format YAML ou JSON. Le modèle CloudFormation décrit les ressources que vous voulez ainsi que leurs paramètres. Utilisez [Compositeur d'infrastructure](#) pour visualiser et valider votre modèle. Cela vous aide à vous assurer que votre modèle est bien structuré et exempt d'erreurs de syntaxe. Pour de plus amples informations, veuillez consulter [Utilisation de CloudFormation modèles](#).
2. Enregistrez le modèle localement ou dans un compartiment Amazon S3.
3. Créez une CloudFormation pile en spécifiant l'emplacement de votre fichier modèle, tel qu'un chemin sur votre ordinateur local ou une URL Amazon S3. Si le modèle contient des paramètres, vous pouvez spécifier des valeurs d'entrée lorsque vous créez la pile. Les paramètres vous permettent de transmettre des valeurs à votre modèle afin de permettre la personnalisation des ressources chaque fois que vous créez une pile.

Note

Si vous spécifiez un fichier modèle stocké localement, CloudFormation il est chargé dans un compartiment S3 de votre AWS compte. CloudFormation crée un compartiment

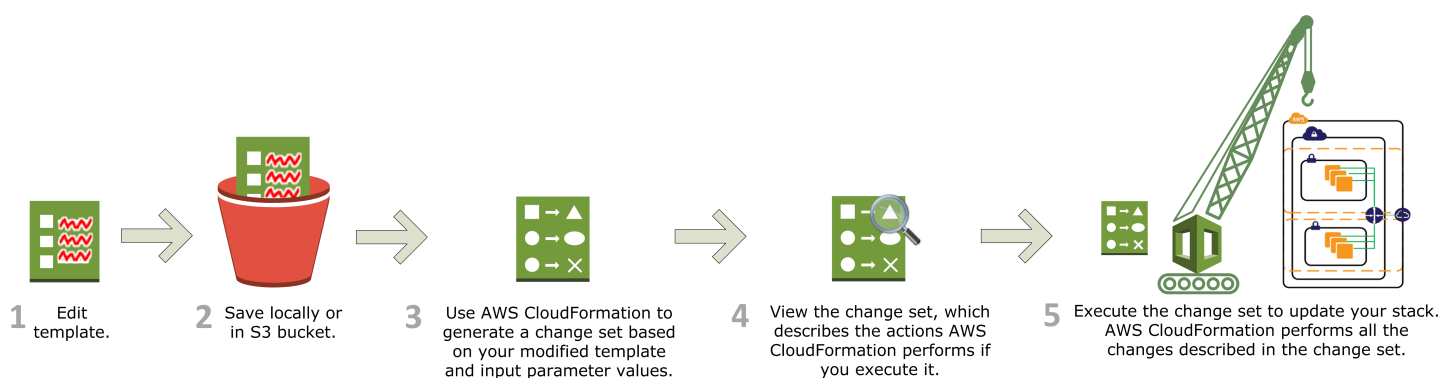
pour chaque région dans laquelle vous téléchargez un fichier modèle. Les buckets sont accessibles à toute personne disposant des autorisations Amazon Simple Storage Service (Amazon S3) sur votre compte. AWS Si un bucket créé par CloudFormation est déjà présent, le modèle est ajouté à ce bucket.

Pour utiliser votre propre compartiment et gérer ses autorisations, vous pouvez importer manuellement les modèles dans Amazon S3. Ensuite, chaque fois que vous créez ou mettez à jour une pile, spécifiez l'URL Amazon S3 d'un fichier de modèle.

Une fois que toutes les ressources ont été créées, CloudFormation indique que votre pile a été créée. Vous pouvez ainsi commencer à en utiliser les ressources. Si la création de la pile échoue, CloudFormation restaure vos modifications en supprimant les ressources qu'il a créées.

Mise à jour d'une pile avec un jeu de modifications

Lorsque vous devez mettre à jour les ressources d'une pile, vous pouvez modifier le modèle correspondant. Vous n'avez pas besoin de créer une autre pile et de supprimer l'ancienne. Pour mettre à jour une pile, créez un ensemble de modifications en soumettant une version modifiée du modèle de pile d'origine, des valeurs de paramètres d'entrée différentes, ou les deux. CloudFormation compare le modèle modifié avec le modèle d'origine et génère un ensemble de modifications. Ce dernier répertorie les modifications proposées. Après avoir passé en revue les modifications, vous pouvez démarrer le jeu de modifications pour mettre à jour la pile ou vous pouvez en créer un autre. Le schéma suivant résume le flux de travail de mise à jour d'une pile.



Pour mettre à jour une pile avec un jeu de modifications

1. Vous pouvez modifier un modèle de CloudFormation pile à l'aide [Compositeur d'infrastructure](#) d'un éditeur de texte. Pour de plus amples informations, veuillez consulter [Mettre à jour votre modèle de pile](#).

Lorsque vous mettez votre modèle à jour, gardez à l'esprit que certaines mises à jour peuvent provoquer des interruptions. En fonction de la ressource et des propriétés concernées, une mise à jour peut interrompre, voire remplacer, une ressource existante. Pour de plus amples informations, veuillez consulter [Comprendre les comportements de mise à jour des ressources d'une pile](#).

2. Enregistrez le CloudFormation modèle localement ou dans un compartiment S3.
3. Pour créer un jeu de modifications, spécifiez la pile que vous souhaitez mettre à jour et l'emplacement du modèle modifié (par exemple, le chemin d'accès sur votre ordinateur local ou une URL Amazon S3). Pour plus d'informations sur la création de jeux de modifications, consultez [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

 Note

Si vous spécifiez un modèle stocké sur votre ordinateur local, le téléchargement CloudFormation automatiquement dans un compartiment S3 de votre Compte AWS.

4. Consultez l'ensemble de modifications pour vérifier qu'il CloudFormation effectuera les modifications attendues. Par exemple, vérifiez si des ressources critiques CloudFormation seront remplacées. Vous pouvez créer autant de jeux de modifications que nécessaire pour inclure toutes les modifications que vous voulez.

 Important

Les jeux de modification n'indiquent pas si la mise à jour de la pile réussira. Par exemple, un jeu de modifications ne vérifie pas si vous dépassez un [quota](#) défini dans votre compte, si vous mettez à jour une [ressource](#) non compatible avec ce type de modification ou si vous disposez des autorisations suffisantes pour modifier une ressource, ce qui peut entraîner l'échec de la mise à jour de la pile.

5. Initiez l'ensemble de modifications que vous souhaitez appliquer à votre pile. CloudFormation met à jour votre pile en ne mettant à jour que les ressources que vous avez modifiées et indique que votre pile a été mise à jour avec succès. Si les mises à jour de la pile CloudFormation échouent, annulez les modifications pour rétablir le dernier état de fonctionnement connu de la pile.

Façons de démarrer avec CloudFormation

Pour créer une CloudFormation pile Hello World avec la console, consultez [Création de votre première pile](#).

Pour un apprentissage guidé, essayez l'atelier [Mise en route avec CloudFormation](#), qui propose une expérience pratique du développement de modèles.

S'inscrire à un Compte AWS

Lorsque vous vous inscrivez AWS, vous êtes automatiquement Compte AWS inscrit à tous les services de AWS, y compris CloudFormation. Si vous en avez un, passez au sujet suivant. Si vous n'en avez pas, utilisez la procédure suivante pour en créer un.

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

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

Sécurisez votre Utilisateur racine d'un compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

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

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

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

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

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

Attribution d'un accès à d'autres utilisateurs

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

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

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

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

Note

Pour plus d'informations sur la gestion des accès, consultez [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

Création de votre première pile

Cette rubrique vous explique comment créer votre première CloudFormation pile à l'aide du AWS Management Console. En suivant ce didacticiel, vous apprendrez à mettre en service des AWS ressources de base, à surveiller les événements liés à la pile et à générer des résultats.

Dans cet exemple, le CloudFormation modèle est écrit en YAML. YAML est un format lisible par l'homme qui est largement utilisé pour définir l'infrastructure en tant que code. Au fur et à mesure que vous en apprendrez plus CloudFormation, vous pourrez également rencontrer d'autres modèles au format JSON, mais pour ce didacticiel, YAML a été choisi pour sa lisibilité.

Note

CloudFormation est gratuit, mais les ressources Amazon EC2 et Amazon S3 que vous créez vous seront facturées. Toutefois, si vous êtes nouveau dans ce AWS domaine, vous pouvez profiter du [niveau gratuit](#) pour minimiser ou éliminer les coûts au cours de ce processus d'apprentissage.

Rubriques

- [Conditions préalables](#)
- [Création d'une CloudFormation pile avec la console](#)
- [Surveiller la création de la pile](#)
- [Tester le service Web](#)
- [Résolution des problèmes](#)
- [Nettoyage](#)
- [Étapes suivantes](#)

Conditions préalables

- Vous devez avoir accès Compte AWS à un utilisateur ou à un rôle IAM autorisé à utiliser Amazon EC2, CloudFormation Amazon S3 et/ou un accès utilisateur administratif.
- Vous devez disposer d'un cloud privé virtuel (VPC) ayant accès à Internet. Ce modèle de procédure pas-à-pas nécessite un VPC par défaut, qui est automatiquement fourni avec les Comptes AWS plus récentes. Si vous ne disposez pas d'un VPC par défaut ou si celui-ci a été supprimé, consultez la section Dépannage de cette rubrique pour trouver d'autres solutions.

Création d'une CloudFormation pile avec la console

Pour créer une CloudFormation pile Hello World avec la console

1. Ouvrez la [CloudFormation console](#).
2. Sélectionnez Créer une pile.
3. Sur la page Créer une pile, sélectionnez Créer à partir d'Infrastructure Composer, puis Créer dans Infrastructure Composer. Cela vous amène à Infrastructure Composer en mode CloudFormation console où vous pouvez télécharger et valider l'exemple de modèle.
4. Pour télécharger et valider le modèle d'exemple, procédez comme suit :
 - a. Sélectionnez Modèles. Copiez et collez ensuite le CloudFormation modèle suivant dans l'éditeur de modèles :

```
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for WebServer with Security Group and EC2
Instance
```

Parameters:**LatestAmiId:**

Description: The latest Amazon Linux 2 AMI from the Parameter Store

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.micro

AllowedValues:

- t3.micro
- t2.micro

ConstraintDescription: must be a valid EC2 instance type.

MyIP:

Description: Your IP address in CIDR format (e.g. 203.0.113.1/32).

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:**WebServerSecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP access via my IP address

SecurityGroupIngress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- CidrIp: !Ref MyIP

WebServer:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData: !Base64 |

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html
```

Outputs:**WebsiteURL:**

Value: !Join

- ''

- - http://

- !GetAtt WebServer.PublicDnsName

Description: Website URL

Avant de passer à l'étape suivante, prenons un moment pour examiner le modèle et comprendre certains CloudFormation concepts clés.

- Cette section **Parameters** déclare les valeurs qui peuvent être transmises au modèle lorsque vous créez la pile. Les ressources spécifiées ultérieurement dans le modèle font référence à ces valeurs et utilisent les données. Les paramètres constituent un moyen efficace de spécifier les informations que vous ne souhaitez pas stocker dans le modèle lui-même. Ils permettent également de spécifier des informations qui peuvent être propres à l'application ou à la configuration spécifique que vous déployez.
- Le modèle définit les paramètres suivants :
 - **LatestAmiId** : récupère le dernier ID AMI Amazon Linux 2 dans le Parameter Store AWS Systems Manager .
 - **InstanceType** : permet de sélectionner le type d'instance EC2 (par défaut : t2.micro, autorisé : t3.micro, t2.micro).
 - **MyIP** : spécifie la plage d'adresses IP pour l'accès HTTP (par défaut : 0.0.0.0/0, permettant l'accès à partir de n'importe quelle adresse IP).
- La **Resources** section contient les définitions des AWS ressources que vous souhaitez créer avec le modèle. Les déclarations de ressource sont un moyen efficace de spécifier tous ces paramètres de configuration simultanément. Lorsque vous incluez des déclarations de ressources dans un modèle, vous pouvez créer et configurer toutes les ressources déclarées en utilisant ce modèle pour créer une pile. Vous pouvez également

créer de nouvelles piles à partir du même modèle pour lancer des configurations identiques de ressources.

- Ce modèle crée les ressources suivantes :
 - **WebServerSecurityGroup** : un groupe de sécurité EC2 qui autorise le trafic HTTP entrant sur le port 80 à partir de la plage d'adresses IP spécifiée.
 - **WebServer** : une instance EC2 avec la configuration suivante :
 - Utilise la dernière AMI Amazon Linux 2
 - Applique le type d'instance sélectionné
 - Ajoute le WebServerSecurityGroup à la propriété SecurityGroupIds
 - Inclut un script de données utilisateur pour installer le serveur HTTP Apache
 - Un nom logique est spécifié au début de chaque déclaration de ressource et de paramètre. Par exemple, WebServerSecurityGroup est le nom logique attribué à la ressource du groupe de sécurité EC2. La fonction Ref est ensuite utilisée pour référencer les ressources et les paramètres par leur nom logique dans d'autres parties du modèle. Lorsqu'une ressource fait référence à une autre ressource, cela crée une dépendance entre elles.
 - La section **Outputs** définit les valeurs personnalisées qui sont renvoyées après la création de la pile. Vous pouvez utiliser les valeurs de sortie pour renvoyer des informations provenant des ressources de la pile, telles que les identificateurs de ressources ou URLs.
 - Le modèle définit une sortie :
 - **WebsiteURL** : l'URL du serveur Web déployé, construite à l'aide du nom DNS public de l'instance EC2. La fonction Join permet de combiner le fixe http:// avec la variable PublicDnsName en une seule chaîne, ce qui facilite la sortie de l'URL complète du serveur Web.
- b. Choisissez Valider pour vous assurer que le code YAML est valide avant de télécharger le modèle.
 - c. Ensuite, sélectionnez Créer un modèle pour créer le modèle et l'ajouter à un compartiment S3.
 - d. Dans la boîte de dialogue qui s'ouvre, notez le nom du compartiment S3 afin de pouvoir le supprimer ultérieurement. Choisissez ensuite Confirmer et passez à CloudFormation. Cela vous amène à la CloudFormation console où le chemin S3 vers votre modèle est désormais spécifié.

5. Sur la page Créer une pile choisissez Suivant.
6. Sur la page Spécifier les détails de la pile, saisissez un nom dans le champ Nom de la pile. Le nom de la pile ne peut pas contenir d'espaces. Pour cet exemple, utilisez **MyTestStack**.
7. Sous Paramètres, spécifiez les valeurs des paramètres comme suit :
 - LatestAmild: Ce paramètre est défini par défaut sur la dernière AMI Amazon Linux 2.
 - InstanceType: Choisissez t2.micro ou t3.micro pour le type d'instance EC2.

Note

Si vous êtes nouveau dans ce AWS domaine, vous pouvez utiliser le niveau gratuit pour lancer et utiliser une t2.micro instance gratuitement pendant 12 mois (dans les régions où elle n'est pas disponible, vous pouvez utiliser une t3.micro instance dans le cadre du niveau gratuit).

- MyIP : Spécifiez votre adresse IP publique réelle à l'aide d'un suffixe /32. Le suffixe /32 est utilisé dans la notation CIDR pour spécifier qu'une seule adresse IP est autorisée. Cela signifie essentiellement autoriser le trafic vers et depuis cette adresse IP spécifique, et aucune autre.
8. Sélectionnez deux fois Suivant pour accéder à la page Vérifier et créer. Pour ce didacticiel, vous pouvez laisser les valeurs par défaut de la page Configurer les options de pile telles quelles.
 9. Vérifiez les informations de la pile. Lorsque vous êtes satisfait de vos paramètres, choisissez Submit (Envoyer).

Surveiller la création de la pile

Après avoir choisi Soumettre, CloudFormation commence à créer les ressources spécifiées dans le modèle. Votre nouvelle pile apparaît dans la liste en haut de la CloudFormationconsole. **MyTestStack** Son statut devrait être CREATE_IN_PROGRESS. Pour voir l'état détaillé d'une pile, consultez ses événements.

Pour afficher les événements de la pile

1. Sur la CloudFormation console, choisissez la pile **MyTestStack** dans la liste.
2. Dans le volet de détails de la pile, choisissez l'onglet Events (Événements).

La console actualise automatiquement la liste en y ajoutant les événements les plus récents toutes les 60 secondes.

L'onglet Événements affiche chaque étape majeure de création de la pile, triée en fonction de l'heure de chaque événement, avec les derniers événements en tête de liste.

Le premier événement (en bas de la liste d'événements) correspond au début du processus de création de la pile :

```
2024-12-23 18:54 UTC-7 MyTestStack CREATE_IN_PROGRESS User initiated
```

Ensuite viennent les événements qui marquent le début et la fin de la création de chaque ressource. Par exemple, la création de l'instance EC2 génère les entrées suivantes :

```
2024-12-23 18:59 UTC-7 WebServer CREATE_COMPLETE
```

```
2024-12-23 18:54 UTC-7 WebServer CREATE_IN_PROGRESS Resource creation initiated
```

L'`CREATE_IN_PROGRESS` événement est enregistré lorsqu'il CloudFormation est indiqué qu'il a commencé à créer la ressource. L'événement `CREATE_COMPLETE` est enregistré lorsque la ressource a été créée avec succès.

Une CloudFormation fois la pile créée avec succès, vous verrez l'événement suivant en haut de l'onglet Événements :

```
2024-12-23 19:17 UTC-7 MyTestStack CREATE_COMPLETE
```

S'il n'est pas CloudFormation possible de créer une ressource, il signale un `CREATE_FAILED` événement et, par défaut, annule la pile et supprime toutes les ressources créées. La colonne Motif du statut affiche le problème qui a provoqué l'échec.

Une fois la pile créée, vous pouvez accéder à l'onglet Ressources pour afficher l'instance EC2 et le groupe de sécurité que vous avez créés.

Tester le service Web

Une fois la pile créée avec succès, accédez à l'onglet Sorties de la CloudFormation console. Recherchez le champ WebsiteURL. Cela contiendra l'URL publique de votre instance EC2.

Ouvrez un navigateur et accédez à l'URL répertoriée sous WebsiteURL. Vous devriez voir un message "Hello World!" simple s'afficher dans le navigateur.

Cela confirme que votre instance EC2 exécute le serveur HTTP Apache et diffuse une page Web de base.

Résolution des problèmes

Si vous rencontrez un retour en arrière lors de la création de la pile, cela peut être dû à l'absence d'un VPC. Voici comment résoudre ce problème.

Aucun VPC par défaut n'est disponible

Le modèle présenté dans cette procédure pas à pas nécessite un VPC par défaut. Si la création de votre pile échoue en raison d'erreurs de disponibilité du VPC ou du sous-réseau, il se peut que vous n'ayez pas de VPC par défaut dans votre compte. Vous avez les options suivantes :

- Créer un nouveau VPC par défaut : vous pouvez créer un nouveau VPC par défaut via la console Amazon VPC. Pour obtenir des instructions, consultez la section [Créer un VPC](#) par défaut dans le Guide de l'utilisateur Amazon VPC.
- Modifier le modèle pour spécifier un sous-réseau — Si vous avez un VPC autre que celui par défaut, vous pouvez modifier le modèle pour spécifier explicitement le VPC et le sous-réseau. IDs Ajoutez le paramètre suivant au modèle :

```
SubnetId:
  Description: The subnet ID to launch the instance into
  Type: AWS::EC2::Subnet::Id
```

Ensuite, mettez à jour la ressource `WebServer` pour inclure l'ID du sous-réseau :

```
WebServer:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    SubnetId: !Ref SubnetId
    UserData: !Base64 |
      #!/bin/bash
      yum update -y
      yum install -y httpd
      systemctl start httpd
      systemctl enable httpd
      echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html
```

Lors de la création de la pile, vous devez spécifier un sous-réseau ayant accès à Internet pour que le serveur Web soit accessible.

Nettoyage

Pour vous assurer que vous n'êtes pas facturé pour des services indésirables, supprimez la pile et ses ressources. Vous pouvez également supprimer le compartiment S3 qui stocke le modèle de la pile.

Pour supprimer la pile et ses ressources

1. Ouvrez la [CloudFormation console](#).
2. Sur la page Piles, sélectionnez l'option à côté du nom de la pile que vous avez créée (**MyTestStack**), puis choisissez Supprimer.
3. Lorsque vous êtes invité à confirmer l'opération, choisissez Supprimer.
4. Surveillez la progression du processus de suppression de la pile dans l'onglet Événement. L'état de **MyTestStack** est remplacé par DELETE_IN_PROGRESS. Lorsque la suppression de la pile est CloudFormation terminée, elle supprime la pile de la liste.

Si vous avez terminé d'utiliser le modèle d'exemple et que vous n'avez plus besoin de votre compartiment Amazon S3, supprimez-le. Avant de pouvoir supprimer un compartiment, vous devez d'abord supprimer son contenu. Vider un compartiment supprime tous les objets qu'il contient.

Pour vider et supprimer le compartiment Amazon S3

1. Ouvrez la [console Amazon S3](#).
2. Dans le volet de navigation situé sur le côté gauche de la console, choisissez Compartiments.
3. Dans la liste Compartiments, sélectionnez l'option à côté du nom du compartiment que vous avez créé pour ce didacticiel, puis choisissez Vider.
4. Sur la page Vider le compartiment, confirmez que vous souhaitez vider le compartiment en saisissant **permanently delete** dans le champ de texte, puis choisissez Vider.
5. Surveillez la progression du processus de vidage du compartiment sur la page Vider le compartiment : état.
6. Pour revenir à votre liste de compartiments, choisissez Quitter.
7. Sélectionnez l'option en regard du nom du compartiment, puis choisissez Supprimer.

8. Lorsque vous êtes invité à confirmer, saisissez le nom du compartiment, puis sélectionnez Supprimer le compartiment.
9. Surveillez la progression du processus de suppression des compartiments à partir de la liste Compartiments. Lorsque Amazon S3 a terminé la suppression du compartiment, il le supprime de la liste.

Étapes suivantes

Félicitations ! Vous avez créé une pile avec succès, surveillé ses créations et utilisé ses résultats.

Pour continuer à apprendre :

- Apprenez-en davantage sur les modèles afin de pouvoir créer vos propres modèles. Pour de plus amples informations, veuillez consulter [Utilisation de CloudFormation modèles](#).
- Essayez l'atelier [Premiers pas avec CloudFormation](#) pour vous exercer davantage à la création de modèles.
- Pour une version abrégée de [Getting Started with CloudFormation](#), voir [Déploiement d'applications sur Amazon EC2](#). Cette rubrique décrit le même scénario d'utilisation d'un script CloudFormation d'assistance pour démarrer une instance Amazon EC2. `cfn-init`

CloudFormation meilleures pratiques

Les meilleures pratiques sont des recommandations qui peuvent vous aider à l'utiliser CloudFormation plus efficacement et à adopter des pratiques sûres tout au long de son flux de travail. Découvrez comment planifier et organiser vos piles, comment créer des modèles qui décrivent vos ressources et les applications qui y sont exécutées, et comment gérer vos piles et leurs ressources. Les meilleures pratiques suivantes sont basées sur l'expérience réelle des CloudFormation clients actuels.

Planification et organisation

- [Réduction du cycle de rétroaction pour accélérer le développement](#)
- [Organisation des piles par cycle de vie et propriétaire](#)
- [Utilisation des références inter-piles pour récupérer la valeur d'une sortie exportée par une autre pile](#)
- [Utilisation CloudFormation StackSets pour les déploiements multicomptes et multirégions](#)
- [Réutilisation de modèles pour répliquer des piles dans plusieurs environnements](#)
- [Vérification des quotas pour tous les types de ressource](#)
- [Utiliser des modules pour réutiliser les configurations de ressources](#)
- [Adoption des pratiques d'infrastructure en tant que code](#)

Création de modèles

- [N'intégrez pas d'informations d'identification dans vos modèles](#)
- [Utiliser des AWS types de paramètres spécifiques](#)
- [Utilisation de contraintes de paramètre](#)
- [Utiliser des pseudoparamètres pour promouvoir la portabilité](#)
- [Utilisation d'AWS::CloudFormation::Init pour déployer des applications dans des instances Amazon Amazon EC2](#)
- [Utilisation des derniers scripts d'assistant](#)
- [Validation des modèles avant leur utilisation](#)
- [Utilisation de YAML ou JSON pour la création de modèles](#)
- [Mise en œuvre d'une stratégie de balisage complète](#)
- [Exploitation des macros de modèle pour des transformations avancées](#)

Gestion des piles

- [Gérez toutes les ressources de la pile via CloudFormation](#)
- [Création de jeux de modification avant la mise à jour des piles](#)
- [Utilisation des politiques de pile pour protéger les ressources](#)
- [AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels](#)
- [Utilisation des révisions de code et des contrôles de révision pour gérer les modèles](#)
- [Mettez à jour régulièrement vos instances Amazon EC2](#)
- [Utilisation régulière de la détection de dérive](#)
- [Configuration des déclencheurs de restauration automatique](#)
- [Mise en œuvre de stratégies efficaces de refactorisation des piles](#)
- [Utilisez les CloudFormation Hooks pour la gestion du cycle de vie](#)

Outils de création

- [Utilisation du générateur IaC pour créer des modèles à partir de ressources existantes](#)
- [Utilisation AWS Infrastructure Composer pour la conception de modèles visuels](#)
- [Envisagez de l'utiliser AWS Cloud Development Kit \(AWS CDK\) pour une infrastructure complexe](#)

Conformité et sécurité

- [Utilisation du contrôle d'accès IAM](#)
- [Application du principe du moindre privilège](#)
- [Sécurisation des paramètres sensibles](#)
- [Implémentez la politique sous forme de code avec AWS CloudFormation Guard](#)

Réduction du cycle de rétroaction pour accélérer le développement

Adoptez des pratiques et des outils qui vous aident à raccourcir la boucle de rétroaction pour votre infrastructure que vous décrivez à l'aide CloudFormation de modèles. Il s'agit notamment de procéder à un linting et à des tests précoces de vos modèles sur votre poste de travail ; vous pouvez ainsi détecter d'éventuels problèmes de syntaxe et de configuration avant même de soumettre vos contributions à un référentiel de code source. La détection précoce de ces problèmes permet d'éviter qu'ils atteignent les environnements formels du cycle de vie, tels que le développement, l'assurance qualité et la production. Cette approche de test précoce et d'interruption immédiate vous permet

de réduire le temps d'attente pour le remaniement, de réduire les zones d'impact potentielles et d'accroître votre niveau de confiance dans la réussite des opérations de dimensionnement.

Parmi les outils qui vous aident à adopter des pratiques rapides, citons [CloudFormation Linter](#) (`cfn-lint`) et [TaskCat](#) les outils de ligne de commande. L'`cfn-lint` outil vous permet de valider vos CloudFormation modèles par rapport à la [spécification CloudFormation des ressources](#). Il s'agit notamment de vérifier les valeurs valides des propriétés des ressources, ainsi que les meilleures pratiques. Des plug-ins pour `cfn-lint` sont [disponibles pour plusieurs éditeurs de code](#), ce qui vous permet de visualiser les problèmes directement dans votre éditeur et d'obtenir un retour de validation immédiat. Vous pouvez également intégrer `cfn-lint` dans la configuration du référentiel de votre code source afin d'exécuter la validation de modèles lors de la validation de vos contributions. Pour plus d'informations, consultez la section [Validation préalable des CloudFormation modèles par Git avec `cfn-lint`](#). Une fois que vous avez effectué le découpage initial et que vous avez résolu les problèmes `cfn-lint` éventuels, vous pouvez tester vos modèles en créant des TaskCat piles par programmation dans le fichier de votre choix. Régions AWS TaskCat génère également un rapport avec des pass/fail notes pour chaque région que vous avez choisie.

[Pour une step-by-step présentation pratique de la façon d'utiliser les deux outils pour raccourcir la boucle de rétroaction, suivez le laboratoire de peluchage et de test de l'CloudFormation atelier.](#)

Organisation des piles par cycle de vie et propriétaire

Utilisez le cycle de vie et la propriété de vos AWS ressources pour vous aider à décider quelles ressources doivent figurer dans chaque pile. Au départ, une pile peut suffire à contenir toutes vos ressources. Toutefois, à mesure que votre pile s'agrandit et se diversifie, sa gestion peut devenir longue et compliquée. En regroupant les ressources par cycle de vie et par propriétaire, les utilisateurs peuvent apporter des modifications à leur ensemble de ressources avec leurs propres processus et leur planification sans affecter les autres ressources.

Par exemple, imaginez une équipe de développeurs et d'ingénieurs responsable d'un site web hébergé sur des instances Amazon EC2 Auto Scaling derrière un équilibreur de charge. Comme le site web a son propre cycle de vie et qu'il est géré par l'équipe web, vous pouvez créer une pile pour le site web et ses ressources. Imaginez maintenant que le site web utilise également des bases de données principales, qui se trouvent dans une pile distincte où elles sont gérées par les administrateurs de base de données qui en sont également les propriétaires. Chaque fois que les membres de l'équipe web ou de l'équipe de base de données doivent mettre à jour leurs ressources, ils peuvent le faire sans affecter la pile de l'autre équipe. Si toutes les ressources étaient dans une seule pile, la coordination et la communication des mises à jour pourraient être difficiles.

Pour plus d'informations sur l'organisation de vos piles, vous pouvez utiliser deux structures communes : une architecture multicouche et une architecture orientée services (SOA).

Une architecture multicouche organise les piles dans plusieurs couches horizontales qui se chevauchent et où chaque couche dépend de la couche sous-jacente. Vous pouvez avoir une ou plusieurs piles dans chaque couche. Toutefois, les ressources AWS des piles de chaque couche doivent avoir des cycles de vie et un propriétaire similaires.

Avec une architecture orientée services, vous pouvez organiser les problématiques stratégiques d'une entreprise en portions gérables. Chacune de ces portions est un service dont l'objectif est clairement défini et qui représente une unité autonome de fonctionnalités. Vous pouvez mapper ces services avec des piles, où chaque pile a son propre cycle de vie et ses propriétaires. Ces services (piles) peuvent être reliés pour qu'ils puissent interagir les uns avec les autres.

Utilisation des références inter-piles pour récupérer la valeur d'une sortie exportée par une autre pile

Lorsque vous organisez vos AWS ressources en fonction du cycle de vie et de la propriété, vous souhaitez peut-être créer une pile qui utilise des ressources se trouvant dans une autre pile. Vous pouvez coder des valeurs en dur ou utiliser des paramètres d'entrée pour transmettre les noms des ressources et IDs. Toutefois, ces méthodes peuvent rendre difficile la réutilisation de modèles ou accroître les traitements nécessaires à l'exécution d'une pile. Au lieu de cela, utilisez des références inter-piles pour récupérer la valeur d'une sortie exportée par une autre pile, afin que d'autres piles puissent l'utiliser. Les piles peuvent utiliser les ressources exportées en les appelant à l'aide de la fonction `Fn::ImportValue`.

Par exemple, si vous disposez d'une pile de réseau constituée d'un VPC, d'un groupe de sécurité et d'un sous-réseau, et que vous souhaitez que toutes les applications web publiques utilisent ces ressources, vous devez exporter ces dernières. Toutes les piles dotées d'applications web publiques pourront ainsi les utiliser. Pour de plus amples informations, veuillez consulter [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#).

Utilisation CloudFormation StackSets pour les déploiements multicomptes et multirégions

CloudFormation StackSets étendez les fonctionnalités des piles en vous permettant de créer, de mettre à jour ou de supprimer des piles sur plusieurs comptes et régions en une seule opération.

StackSets À utiliser pour déployer des composants d'infrastructure communs, des contrôles de conformité ou des services partagés au sein de votre organisation.

Lors de l'utilisation StackSets, implémentez les autorisations gérées par les services AWS Organizations pour simplifier la gestion des autorisations. Cette approche vous permet de déployer sur des comptes StackSets au sein de votre organisation sans avoir à configurer manuellement les rôles IAM dans chaque compte.

Pour plus d'informations sur la StackSets section [StackSets Concepts](#).

Vérification des quotas pour tous les types de ressource

Avant de lancer une pile, assurez-vous de pouvoir créer toutes les ressources que vous souhaitez sans atteindre les limites de votre AWS compte. Si vous atteignez une limite, vous ne CloudFormation créez pas votre pile avec succès tant que vous n'aurez pas augmenté votre quota ou supprimé des ressources supplémentaires. Chaque service peut imposer des limites différentes que vous devez connaître avant de lancer une pile. Par exemple, par défaut, vous ne pouvez lancer que 2 000 CloudFormation piles par région dans votre Compte AWS. Pour plus d'informations sur les limites et sur l'augmentation des limites par défaut, consultez [service quotas AWS](#) dans Références générales AWS.

Réutilisation de modèles pour répliquer des piles dans plusieurs environnements

Une fois que les piles et les ressources sont configurées, vous pouvez réutiliser vos modèles afin de répliquer votre infrastructure dans plusieurs environnements. Par exemple, vous pouvez créer des environnements de développement, de test et de production afin de pouvoir tester les modifications avant de les migrer en production. Pour que les modèles soient réutilisables, recourez aux sections Parameters, Mappings et Conditions afin de personnaliser les piles lorsque vous les créez. Par exemple, pour vos environnements de développement, vous pouvez spécifier un type d'instance qui coûte moins cher que votre environnement de production, mais où toutes les autres configurations et tous les autres paramètres restent les mêmes. Pour plus d'informations sur les paramètres, les mappages et les conditions, consultez [CloudFormation sections du modèle](#).

Utiliser des modules pour réutiliser les configurations de ressources

A mesure que votre infrastructure s'agrandit, des tendances peuvent émerger où vous déclarez les mêmes composants dans chacun de vos modèles. Les modules vous permettent d'empaqueter des configurations de ressources pour les inclure dans les modèles de pile, de manière transparente, gérable et reproductible. Les modules peuvent encapsuler les configurations de service courantes et les bonnes pratiques en tant que blocs de création modulaires et personnalisables à inclure dans vos modèles de pile.

Ces blocs de construction peuvent être destinés à une seule ressource, telle que les bonnes pratiques pour définir une instance Amazon Elastic Compute Cloud (Amazon EC2), ou à plusieurs ressources pour définir des modèles communs d'architecture d'application. Ces éléments de base peuvent être imbriqués dans d'autres modules, afin que vous puissiez regrouper vos meilleures pratiques dans des éléments de base de niveau supérieur. CloudFormation les modules sont disponibles dans le [CloudFormation registre](#), vous pouvez donc les utiliser comme une ressource native. Lorsque vous utilisez un CloudFormation module, le modèle de module est développé dans le modèle de consommation, ce qui vous permet d'accéder aux ressources du module à l'aide d'un [Ref](#) ou d'un [Fn::GetAtt](#). Pour de plus amples informations, veuillez consulter [Créez des configurations de ressources réutilisables qui peuvent être incluses dans les modèles à l'aide de CloudFormation modules](#).

Adoption des pratiques d'infrastructure en tant que code

Traitez vos CloudFormation modèles comme du code en mettant en œuvre des pratiques d'infrastructure en tant que code (IaC). Stockez vos modèles dans des systèmes de gestion des versions, mettez en œuvre des révisions de code et utilisez des tests automatisés pour valider les modifications. Cette approche garantit la cohérence, améliore la collaboration et fournit une piste d'audit pour les modifications d'infrastructure.

Envisagez d'implémenter des CI/CD pipelines pour votre code d'infrastructure afin d'automatiser les tests et le déploiement de vos CloudFormation modèles. Des outils tels que AWS CodePipeline, AWS CodeBuild, et AWS CodeDeploy peuvent être utilisés pour créer des flux de travail automatisés pour les déploiements de votre infrastructure.

Pour plus d'informations sur la mise en œuvre des meilleures pratiques IaC, voir [Utilisation en AWS CloudFormation tant qu'outil IaC](#).

Pour plus d'informations sur l'utilisation de la livraison continue avec CloudFormation, consultez la section [Livraison continue avec CodePipeline](#).

N'intégrez pas d'informations d'identification dans vos modèles

Plutôt que d'intégrer des informations sensibles dans vos CloudFormation modèles, nous vous recommandons d'utiliser des références dynamiques dans votre modèle de pile.

Les références dynamiques constituent un moyen compact et puissant de référencer des valeurs externes stockées et gérées dans d'autres services, tels que le magasin de paramètres de AWS Systems Manager ou le gestionnaire de AWS secrets. Lorsque vous utilisez une référence dynamique, CloudFormation récupère la valeur de la référence spécifiée lorsque cela est nécessaire lors des opérations de pile et de modification des ensembles de modifications, et transmet la valeur à la ressource appropriée. Cependant, CloudFormation ne stocke jamais la valeur de référence réelle. Pour de plus amples informations, veuillez consulter [Utilisation de références dynamiques pour spécifier des valeurs de modèle](#).

[AWS Secrets Manager](#) vous aide à chiffrer, stocker et récupérer en toute sécurité les informations d'identification pour vos bases de données et autres services. [AWS Systems Manager Parameter Store](#) offre un stockage sécurisé et hiérarchique de la gestion des données de configuration.

Pour plus d'informations sur la définition des paramètres de modèle, consultez [CloudFormation Parameters syntaxe du modèle](#).

Utiliser des AWS types de paramètres spécifiques

Si votre modèle nécessite des entrées pour AWS des valeurs spécifiques existantes, telles que des identifiants Amazon Virtual Private Cloud existants ou un nom de paire de clés Amazon EC2, AWS utilisez des types de paramètres spécifiques. Par exemple, vous pouvez spécifier un paramètre en tant que `typeAWS::EC2::KeyPair::KeyName`, qui prend le nom d'une paire de clés existante qui se trouve dans votre région Compte AWS et dans la région dans laquelle vous créez la pile. CloudFormation peut rapidement valider les valeurs pour des types de paramètres AWS spécifiques avant de créer votre pile. De plus, si vous utilisez la CloudFormation console, elle CloudFormation affiche une liste déroulante de valeurs valides, de sorte que vous n'avez pas à rechercher ou à mémoriser les noms de VPC IDs ou de paires de clés corrects. Pour de plus amples informations, veuillez consulter [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#).

Utilisation de contraintes de paramètre

Avec les contraintes, vous pouvez décrire les valeurs d'entrée autorisées afin CloudFormation de détecter toute valeur non valide avant de créer une pile. Vous pouvez définir des contraintes comme une longueur minimale, une longueur maximale et des modèles autorisés. Par exemple, vous pouvez définir des contraintes au niveau de la valeur du nom d'utilisateur d'une base de données, de sorte qu'elle contienne au moins huit caractères alphanumériques. Pour de plus amples informations, veuillez consulter [CloudFormation Parameters syntaxe du modèle](#).

Utiliser des pseudoparamètres pour promouvoir la portabilité

Vous pouvez utiliser des [pseudo-paramètres](#) dans vos modèles comme arguments pour des [fonctions intrinsèques](#), telles que Ref et Fn::Sub. Les pseudo-paramètres sont des paramètres prédéfinis par CloudFormation. Vous ne les déclarez pas dans le modèle. L'utilisation de pseudo-paramètres dans les fonctions intrinsèques augmente la portabilité de vos modèles de pile entre les régions et les comptes.

Par exemple, imaginez que vous souhaitiez créer un modèle dans lequel, pour une propriété de ressource donnée, vous devez spécifier l'[Amazon Resource Name](#) (ARN) d'une autre ressource existante. Dans ce cas, la ressource existante est une ressource [AWS Systems Manager Parameter Store](#) avec l'ARN suivant : `arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter`. Vous devrez adapter le [format ARN](#) à votre partition cible AWS, à votre région et à votre ID de compte. Au lieu de coder ces valeurs en dur, vous pouvez utiliser des pseudo-paramètres `AWS::Partition`, `AWS::Region` et `AWS::AccountId` pour rendre votre modèle plus portable. Dans ce cas, l'exemple suivant vous montre comment concaténer des éléments d'un ARN avec : CloudFormation !Sub 'arn:\${AWS::Partition}:ssm:\${AWS::Region}:\${AWS::AccountId}:parameter/MySampleParameter

Pour un autre exemple, supposons que vous souhaitiez partager des ressources ou des configurations entre plusieurs piles. Dans cet exemple, supposons que vous avez créé un [sous-réseau](#) pour votre VPC, puis exporté son ID pour l'utiliser avec d'autres piles du même VPC et de la Compte AWS même région. Dans une autre pile, vous faites référence à la valeur exportée de l'ID de sous-réseau lorsque vous décrivez une instance Amazon EC2. Pour un exemple détaillé de l'utilisation du champ de sortie Export et de la fonction intrinsèque Fn::ImportValue, consultez [Référez-vous aux sorties de ressources dans une autre CloudFormation pile](#).

Les exportations de pile doivent être uniques par compte et par région. Dans ce cas, vous pouvez donc utiliser le pseudo paramètre `AWS::StackName` pour créer un préfixe pour votre exportation.

Comme les noms de pile doivent également être uniques par compte et par région, l'utilisation de ce pseudo-paramètre comme préfixe augmente la possibilité d'avoir un nom d'exportation unique, tout en promouvant une approche réutilisable dans les piles à partir desquelles vous exportez des valeurs. Vous pouvez également utiliser un préfixe de votre choix.

Utilisation d'**AWS::CloudFormation::Init** pour déployer des applications dans des instances Amazon Amazon EC2

Lorsque vous lancez des piles, vous pouvez installer et configurer des applications logicielles sur des instances Amazon EC2 via le script d'assistance `cfn-init` et la ressource `AWS::CloudFormation::Init`. Avec `AWS::CloudFormation::Init`, vous pouvez décrire les configurations de votre choix au lieu de créer des scripts pour chaque étape de la procédure. Vous pouvez également mettre à jour les configurations sans avoir à recréer les instances. Et en cas de problème avec votre configuration, CloudFormation génère des journaux que vous pouvez utiliser pour étudier les problèmes.

Dans le modèle, spécifiez les états de l'installation et de la configuration dans la ressource `AWS::CloudFormation::Init`. Pour la procédure pas à pas montrant comment utiliser `cfn-init` et `AWS::CloudFormation::Init`, consultez [Déploiement d'applications sur Amazon EC2](#).

Utilisation des derniers scripts d'assistant

Les scripts CloudFormation d'assistance sont régulièrement mis à jour. Veillez à inclure la commande suivante dans la propriété `UserData` de votre modèle avant d'appeler les scripts d'assistant pour vous assurer que vos instances lancées obtiennent les derniers scripts d'assistant :

```
yum install -y aws-cfn-bootstrap
```

Pour plus d'informations sur l'obtention des derniers scripts d'assistance, consultez la référence des [scripts d'CloudFormation assistance dans le Guide de référence](#) des CloudFormation modèles.

Validation des modèles avant leur utilisation

Avant d'utiliser un modèle pour créer ou mettre à jour une pile, vous pouvez l'utiliser CloudFormation pour le valider. La validation d'un modèle peut vous aider à détecter les erreurs de syntaxe et certaines erreurs sémantiques, telles que les dépendances circulaires, avant de CloudFormation

créer des ressources. Si vous utilisez la CloudFormation console, celle-ci valide automatiquement le modèle une fois que vous avez spécifié les paramètres d'entrée. Pour l' CloudFormation API AWS CLI or, utilisez la commande [validate-template](#) CLI ou [ValidateTemplate](#) l'opération API.

Lors de la validation, CloudFormation vérifiez d'abord si le modèle est un JSON valide. Si ce n'est pas le cas, CloudFormation vérifie si le modèle est dans un format YAML valide. Si les deux vérifications échouent, CloudFormation renvoie une erreur de validation du modèle.

Valider les modèles pour la conformité des politiques de l'organisation

Vous pouvez également valider la conformité de votre modèle aux directives de la politique de l'organisation. AWS CloudFormation Guard (`cfn-guard`) est un outil d'interface de ligne de commande (CLI) open source qui fournit un policy-as-code langage permettant de définir des règles permettant de vérifier les configurations de ressources requises et interdites. Il vous permet ensuite de valider vos modèles par rapport à ces règles. Par exemple, les administrateurs peuvent créer des règles pour garantir que les utilisateurs créent toujours des compartiments Amazon S3 chiffrés.

Vous pouvez l'utiliser `cfn-guard` soit localement, lors de la modification des modèles, soit automatiquement dans le cadre d'un CI/CD pipeline pour arrêter le déploiement de ressources non conformes.

En outre, `cfn-guard` inclut une fonctionnalité qui vous permet d'extraire des règles à partir de CloudFormation modèles conformes existants. `rulegen`

Pour plus d'informations, consultez le référentiel [cfn-guard](#) sur GitHub

Utilisation de YAML ou JSON pour la création de modèles

CloudFormation prend en charge les formats YAML et JSON pour les modèles. Chaque format présente des avantages, et le choix dépend de vos besoins :

Utilisez YAML lorsque

- Vous privilégiez la lisibilité humaine et la maintenabilité
- Vous souhaitez inclure des commentaires pour documenter votre modèle
- Vous travaillez sur des modèles complexes avec des structures imbriquées
- Vous souhaitez utiliser des caractéristiques propres à YAML, telles que les ancres et les alias, pour réduire la répétition

Utilisez JSON lorsque :

- Vous devez intégrer des outils ou systèmes privilégiant JSON
- Vous travaillez sur la génération ou la manipulation de modèles programmatiques
- Vous avez besoin d'une validation stricte des données

Le format YAML est généralement recommandé pour une création manuelle grâce à sa lisibilité et sa prise en charge des commentaires. Il est particulièrement utile pour les modèles complexes où l'indentation facilite la visualisation des hiérarchies de ressources. Le JSON peut être avantageux dans les flux de travail automatisés ou lorsque vous travaillez avec APIs des entrées JSON attendues. C'est également avantageux lorsque vous devez garantir le strict respect d'une structure spécifique. Quel que soit le format choisi, privilégiez des modèles bien structurés, documentés et maintenables. Si vous utilisez le format YAML, tirez parti de ses fonctionnalités telles que les ancres et les alias pour réduire les répétitions et améliorer la maintenabilité.

Mise en œuvre d'une stratégie de balisage complète

Mettez en œuvre une stratégie de balisage cohérente pour toutes les ressources créées par vos CloudFormation modèles. Les balises facilitent l'organisation des ressources, la répartition des coûts, le contrôle d'accès et l'automatisation. Envisagez d'inclure des balises pour l'environnement, le propriétaire, le centre de coûts, l'application et l'objectif.

Utilisez la propriété `Tags` de la ressource `AWS::CloudFormation::Stack` pour appliquer des balises à toutes les ressources prises en charge dans une pile. Vous pouvez également utiliser la propriété `TagSpecifications` disponible sur de nombreux types de ressources pour appliquer des balises lors de la création des ressources.

Pour plus d'informations sur le balisage, consultez [Balisage des ressources](#).

Exploitation des macros de modèle pour des transformations avancées

CloudFormation les macros vous permettent d'effectuer un traitement personnalisé sur des modèles, qu'il s'agisse d'actions simples telles que find-and-replace des opérations ou de transformations complexes générant des ressources supplémentaires. Utilisez des macros pour étendre les fonctionnalités des CloudFormation modèles et implémenter des modèles réutilisables au sein de votre organisation.

AWS Serverless Application Model Voici un exemple de macro qui simplifie le développement d'applications sans serveur. Envisagez la création de macros personnalisées adaptées aux modèles, contraintes et exigences propres à votre organisation.

Pour plus d'informations sur l'utilisation des macros dans vos modèles, consultez la section [Présentation des CloudFormation macros](#).

Gérez toutes les ressources de la pile via CloudFormation

Après avoir lancé une pile, utilisez la CloudFormation [console](#), l'[API](#) ou la [AWS CLI](#) pour mettre à jour les ressources de votre pile. N'apportez pas de modifications pour empiler des ressources à l'extérieur de CloudFormation. Cette approche peut générer un décalage entre le modèle de votre pile et l'état actuel de ses ressources, ce qui peut entraîner des erreurs si vous mettez à jour ou supprimez la pile. Ce phénomène est appelé dérive. Si une modification est apportée à une ressource en dehors du CloudFormation modèle et que vous mettez à jour la pile, les modifications apportées directement à la ressource seront supprimées et la configuration de la ressource reviendra à la configuration du modèle.

Pour plus d'informations sur la dérive, consultez [Qu'est-ce qu'un écart ?](#).

Pour plus d'informations sur la mise à jour des piles, consultez [Mettre à jour une CloudFormation pile](#).

Création de jeux de modification avant la mise à jour des piles

Les ensembles de modifications vous permettent de voir comment les modifications proposées à une pile peuvent avoir un impact sur vos ressources courantes avant de les implémenter. CloudFormation n'apporte aucune modification à votre pile tant que vous n'avez pas exécuté l'ensemble de modifications, ce qui vous permet de décider de poursuivre les modifications proposées ou de créer un autre ensemble de modifications.

Utilisez les jeux de modification pour vérifier l'impact possible des modifications sur les ressources exécutées, notamment les ressources critiques. Par exemple, si vous modifiez le nom d'une instance de base de données Amazon RDS, CloudFormation vous créez une nouvelle base de données et supprimez l'ancienne ; vous perdrez les données de l'ancienne base de données à moins que vous ne les ayez déjà sauvegardées. Si vous générez un jeu de modification, vous verrez que la modification remplacera la base de données. Vous pouvez donc vous préparer à la mise à jour la pile en conséquence. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

Utilisation des politiques de pile pour protéger les ressources

Les politiques de pile contribuent à protéger les ressources critiques d'une pile contre toute mise à jour accidentelle qui pourrait entraîner l'interruption des ressources, voire leur remplacement. Une politique de pile est un document JSON qui décrit les actions de mise à jour qui peuvent être effectuées au niveau des ressources désignées. Spécifiez une politique de pile chaque fois que vous créez une pile qui contient des ressources critiques.

Pendant une mise à jour de la pile, vous devez spécifier explicitement les ressources protégées que vous souhaitez mettre à jour. Dans le cas contraire, aucune modification ne sera apportée aux ressources protégées. Pour de plus amples informations, veuillez consulter [Empêcher les mises à jour des ressources de la pile](#).

AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels

AWS CloudTrail suit toute personne effectuant des appels d' CloudFormation API dans votre Compte AWS. Les appels d'API sont enregistrés chaque fois que quelqu'un utilise l' CloudFormation API, la CloudFormation console, une console principale ou des CloudFormation AWS CLI commandes. Activez la journalisation et spécifiez un compartiment Amazon S3 pour y stocker les journaux. Ainsi, si vous en avez besoin, vous pouvez vérifier qui a effectué quel CloudFormation appel sur votre compte.

Pour de plus amples informations, veuillez consulter [Journalisation des appels d' CloudFormation API avec AWS CloudTrail](#).

Utilisation des révisions de code et des contrôles de révision pour gérer les modèles

Vos modèles de pile décrivent la configuration de vos AWS ressources, telles que les valeurs de leurs propriétés. Pour passer en revue les modifications et pour conserver un historique précis de vos ressources, utilisez les révisions de code et les contrôles de révision. Ces méthodes vous permettent de suivre les modifications apportées entre différentes versions de vos modèles, ce qui peut vous aider à identifier les modifications apportées aux ressources de votre pile. En outre, en conservant un historique, vous pouvez toujours restaurer la pile sur la base d'une certaine version de votre modèle.

Mettez à jour régulièrement vos instances Amazon EC2

Sur toutes vos instances Amazon EC2 Windows et les instances Amazon EC2 Linux créées CloudFormation avec, exécutez régulièrement `yum update` la commande pour mettre à jour le package RPM. Cela garantit que vous obtenez les derniers correctifs et mises à jour de sécurité.

Utilisation régulière de la détection de dérive

Utilisez régulièrement la fonction de détection de CloudFormation dérive pour identifier les ressources qui ont été modifiées en dehors de la CloudFormation gestion. La détection et la résolution des dérives contribuent à préserver l'intégrité de votre approche d'infrastructure en tant que code et à garantir que vos modèles reflètent fidèlement l'état de vos ressources déployées.

Envisagez de mettre en œuvre une détection automatique de la dérive dans le cadre de vos procédures opérationnelles. Vous pouvez utiliser AWS Lambda les fonctions déclenchées par EventBridge les règles d'Amazon pour vérifier régulièrement l'absence de dérive et informer votre équipe lorsque des anomalies sont détectées.

Pour plus d'informations sur la dérive, consultez [Détection de modifications non gérées de la configuration des piles et des ressources avec la détection de dérive](#).

Configuration des déclencheurs de restauration automatique

Utilisez des déclencheurs d'annulation pour spécifier les CloudWatch alarmes Amazon qui CloudFormation doivent être surveillées lors des opérations de création et de mise à jour de la pile. Si l'une des alarmes spécifiées passe à l'ALARM état, annule CloudFormation automatiquement l'ensemble du fonctionnement de la pile, afin de garantir la stabilité de votre infrastructure.

Configurez des déclencheurs de restauration pour des métriques critiques comme les taux d'erreur des applications, l'utilisation des ressources du système, ou des métriques commerciales personnalisées qui indiquent l'état de votre application et de votre infrastructure.

Pour plus d'informations sur les déclencheurs de restauration, consultez [Restauration de la pile en cas de déclenchement d'une alarme](#).

Mise en œuvre de stratégies efficaces de refactorisation des piles

Au fur et à mesure que votre infrastructure évolue, vous devrez peut-être refactoriser vos CloudFormation stacks pour améliorer la maintenabilité, réduire la complexité ou vous adapter

à l'évolution des exigences. La refactorisation des piles consiste à réorganiser vos modèles et ressources tout en préservant leur comportement et leur fonctionnement externes. Il est avantageux d'utiliser le refactoring de la pile des CloudFormation manières suivantes :

- Fragmentation des piles monolithiques : diviser de grandes piles complexes en piles plus petites et plus faciles à gérer, organisées par cycle de vie ou responsabilité
- Consolidation des ressources connexes : regrouper dans une seule pile des ressources liées provenant de plusieurs piles afin de simplifier leur gestion
- Extraction de composants réutilisables : déplacer les modèles récurrents dans des modules ou des piles imbriquées pour favoriser la réutilisation et la cohérence
- Amélioration de l'organisation des ressources : réorganiser les ressources d'une pile pour mieux refléter leurs relations et dépendances

Pour plus d'informations sur la refactorisation de vos CloudFormation piles, consultez. [Refactorisation de piles](#)

Utilisez les CloudFormation Hooks pour la gestion du cycle de vie

CloudFormation Les hooks fournissent du code qui inspecte de manière proactive la configuration de vos AWS ressources avant le provisionnement et effectue des contrôles de validation complexes. Les hooks permettent de vérifier la conformité de vos ressources, de vos piles et de vos ensembles de modifications aux exigences de sécurité, d'exploitation et d'optimisation des coûts. Ils génèrent des avertissements avant le provisionnement d'une ressource, ou avant l'échec de l'opération et son arrêt complet, selon la configuration du hook. Les violations et les avertissements sont enregistrés sur Amazon CloudWatch afin de fournir une visibilité sur les déploiements non conformes.

Pour plus d'informations sur les bonnes pratiques relatives aux hooks, consultez [Concepts des hooks AWS CloudFormation](#).

Pour plus d'informations sur ce que les Hooks peuvent faire pour vos CloudFormation ressources, voir [Que sont les AWS CloudFormation Hooks ?](#)

Utilisation du générateur IaC pour créer des modèles à partir de ressources existantes

Le générateur CloudFormation iAc (infrastructure en tant que code) vous aide à créer des CloudFormation modèles à partir de vos AWS ressources existantes. Cette fonctionnalité est

particulièrement utile lorsque vous devez répliquer une infrastructure existante, documenter des ressources créées manuellement ou gérer des ressources non gérées auparavant. CloudFormation Le générateur IaC est utile pour créer vos CloudFormation modèles de la manière suivante :

- Création accélérée de modèles : générer des modèles à partir de ressources existantes au lieu de les écrire à partir de rien
- Infrastructure cohérente : s'assurer que les nouveaux environnements correspondent aux environnements existants en utilisant les modèles générés comme point de départ
- Migration vers l'infrastructure sous forme de code : CloudFormation gestion progressive des ressources créées manuellement
- Documentation : créer un enregistrement de votre infrastructure existante sous forme de modèle

Pour plus d'informations sur le générateur iAC, consultez [Génération de modèles à partir de ressources existantes avec le générateur IaC](#).

Utilisation AWS Infrastructure Composer pour la conception de modèles visuels

AWS Infrastructure Composer est un outil de conception visuelle qui vous permet de créer, de visualiser et de modifier des CloudFormation modèles à l'aide d'une drag-and-drop interface. Cela peut être particulièrement bénéfique lorsqu'il est utilisé de la CloudFormation manière suivante :

- Planification de l'architecture : concevoir et valider des architectures d'infrastructure avant leur mise en œuvre
- Modernisation des modèles : visualiser les modèles existants pour comprendre leur structure et identifier les opportunités d'amélioration
- Formation et intégration : aidez les nouveaux membres de l'équipe à comprendre les CloudFormation concepts et les relations AWS de service grâce à l'apprentissage visuel
- Communication avec les parties prenantes : présenter les conceptions d'infrastructure aux parties prenantes non techniques à l'aide de représentations visuelles claires
- Examens de conformité : utiliser des diagrammes visuels pour faciliter les examens de sécurité et de conformité de vos conceptions d'infrastructure
- Examens de conformité : utiliser des diagrammes visuels pour faciliter les examens de sécurité et de conformité de vos conceptions d'infrastructure

Pour plus d'informations sur Infrastructure Composer, voir [Qu'est-ce que c'est AWS Infrastructure Composer ?](#).

Envisagez de l'utiliser AWS Cloud Development Kit (AWS CDK) pour une infrastructure complexe

Pour les exigences d'infrastructure complexes, pensez à utiliser le CDK pour définir vos ressources cloud à l'aide de langages de programmation TypeScript courants tels que Python, Java et .NET. AWS CDK génère CloudFormation des modèles à partir de votre code, ce qui vous permet de tirer parti de toutes les fonctionnalités CloudFormation tout en utilisant les abstractions et les constructions de programmation de votre langage préféré.

AWS CDK Il fournit des constructions de haut niveau qui encapsulent les meilleures pratiques et simplifient la définition des modèles d'infrastructure courants. Cela peut réduire considérablement la quantité de code nécessaire pour définir votre infrastructure tout en garantissant le respect des bonnes pratiques.

Pour plus d'informations sur le CDK, consultez [AWS Cloud Development Kit \(AWS CDK\)](#).

Utilisation du contrôle d'accès IAM

IAM est un AWS service que vous pouvez utiliser pour gérer les utilisateurs et leurs autorisations. AWS Vous pouvez utiliser IAM with CloudFormation pour spécifier les CloudFormation actions que les utilisateurs peuvent effectuer, telles que l'affichage des modèles de piles, la création de piles ou la suppression de piles. En outre, toute personne qui gère les piles CloudFormation a besoin des autorisations d'accès requises aux ressources de ces piles. Par exemple, si les utilisateurs souhaitent lancer, mettre CloudFormation à jour ou résilier des instances Amazon EC2, ils doivent être autorisés à lancer les actions Amazon EC2 pertinentes.

Dans la plupart des cas, les utilisateurs ont besoin d'un accès complet pour gérer toutes les ressources dans un modèle. CloudFormation effectue des appels pour créer, modifier et supprimer ces ressources en leur nom. Pour séparer les autorisations entre un utilisateur et le CloudFormation service, utilisez un rôle de service. CloudFormation utilise la politique du rôle de service pour effectuer des appels au lieu de la politique de l'utilisateur. Pour de plus amples informations, veuillez consulter [CloudFormation rôle de service](#).

Application du principe du moindre privilège

Lorsque vous configurez des rôles IAM pour des rôles de CloudFormation service ou pour des ressources créées par vos modèles, appliquez toujours le principe du moindre privilège. N'accordez que les autorisations strictement nécessaires à la fonctionnalité prévue et évitez autant que possible les autorisations génériques avec caractères génériques.

Utilisez IAM Access Analyzer pour examiner les autorisations accordées à vos rôles de CloudFormation service et identifier les autorisations non utilisées qui peuvent être supprimées. Examinez et mettez régulièrement à jour les politiques IAM afin de vous assurer qu'elles restent conformes à vos exigences de sécurité.

Sécurisation des paramètres sensibles

Pour les informations sensibles telles que les mots de passe, les clés d'API et autres secrets, utilisez AWS Systems Manager Parameter Store ou AWS Secrets Manager au lieu de les intégrer directement dans vos modèles. Utilisez des références dynamiques dans vos modèles pour extraire ces valeurs de manière sécurisée lors des opérations sur les piles.

Lorsque vous utilisez des paramètres dans vos modèles, définissez la propriété NoEcho sur `true` pour les paramètres sensibles, afin d'éviter que leurs valeurs ne s'affichent dans la console, dans les réponses de l'API ou dans la sortie de la CLI. Notez toutefois que NoEcho n'empêche pas la valeur d'être consignée si elle est transmise à d'autres services ou ressources susceptibles de la journaliser.

Pour plus d'informations sur l'utilisation de AWS Systems Manager Parameter Store avec la CloudFormation section [Obtenir une valeur en texte brut à partir de AWS Systems Manager Parameter Store](#).

Pour plus d'informations sur l'utilisation de la NoEcho propriété, consultez la section [Syntaxe des paramètres du CloudFormation modèle](#).

Pour plus d'informations sur l'utilisation AWS Secrets Manager avec, CloudFormation voir [Créer des AWS Secrets Manager secrets dans AWS CloudFormation](#).

Implémentez la politique sous forme de code avec AWS CloudFormation Guard

AWS CloudFormation Guard (`cfn-guard`) est un policy-as-code outil open source qui vous permet de définir et d'appliquer des règles pour vos CloudFormation modèles. Utilisez `cfn-guard` pour

garantir que vos modèles respectent les politiques organisationnelles, les bonnes pratiques de sécurité et les exigences de gouvernance.

`cfn-guard` Intégrez-les à vos CI/CD pipelines pour valider automatiquement les modèles par rapport à vos règles de politique avant leur déploiement. Cela permet d'empêcher le déploiement de ressources non conformes dans votre environnement et fournit un retour rapide aux développeurs en cas de violation de règles.

Pour plus d'informations sur Guard, voir [Qu'est-ce que c'est AWS CloudFormation Guard ?](#)

Utilisation de CloudFormation modèles

Un AWS CloudFormation modèle définit les AWS ressources que vous souhaitez créer, mettre à jour ou supprimer dans le cadre d'une pile. Il se compose de plusieurs sections, mais la seule section obligatoire est la section [Resources](#), qui doit déclarer au moins une ressource.

Vous pouvez créer des modèles à l'aide des méthodes suivantes :

- AWS Infrastructure Composer : une interface visuelle pour concevoir des modèles.
- Éditeur de texte : écrivez des modèles directement dans la syntaxe JSON ou YAML.
- Générateur IaC : générez des modèles à partir des ressources fournies dans votre compte qui ne sont pas actuellement gérées par CloudFormation. Le générateur IaC fonctionne avec un large éventail de types de ressources pris en charge par l'API Cloud Control dans votre région.

Cette section fournit un guide complet expliquant comment utiliser les différentes sections d'un CloudFormation modèle et comment commencer à créer des modèles de pile. Elle couvre les rubriques suivantes :

Rubriques

- [Où sont stockés les modèles](#)
- [Validation des modèles](#)
- [Premiers pas avec les modèles](#)
- [Exemples de modèle](#)
- [CloudFormation format de modèle](#)
- [CloudFormation sections du modèle](#)
- [Créez des modèles visuellement avec Infrastructure Composer](#)
- [AWS CloudFormation Serveur de langues](#)
- [Générez des modèles à partir de ressources existantes avec le générateur IaC](#)
- [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#)
- [Obtenir des AWS valeurs à l'aide de pseudo-paramètres](#)
- [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#)
- [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#)

- [CloudFormation procédures pas à pas](#)
- [CloudFormation extraits de modèles](#)
- [Déployez Windows des piles basées sur CloudFormation](#)
- [Étendez les fonctionnalités de votre modèle avec les CloudFormation types de ressources fournis](#)
- [Créez des configurations de ressources réutilisables qui peuvent être incluses dans les modèles à l'aide de CloudFormation modules](#)

Où sont stockés les modèles

Compartiment Amazon S3

Vous pouvez stocker CloudFormation des modèles dans un compartiment Amazon S3. Lorsque vous créez ou mettez à jour une pile, vous pouvez spécifier l'URL S3 du modèle au lieu de le charger directement.

Si vous chargez des modèles directement via le AWS Management Console ou AWS CLI, un compartiment S3 est automatiquement créé pour vous. Pour de plus amples informations, veuillez consulter [Création d'une pile depuis la CloudFormation console](#).

Référentiel Git

Avec [Git sync](#), vous pouvez stocker des modèles dans un référentiel Git. Lorsque vous créez ou mettez à jour une pile, vous pouvez spécifier l'emplacement du dépôt Git et la branche contenant le modèle au lieu de le télécharger directement ou de faire référence à une URL S3. CloudFormation surveille automatiquement le référentiel et la branche spécifiés pour détecter les modifications du modèle. Pour de plus amples informations, veuillez consulter [Créez une pile à partir du code source du référentiel avec la synchronisation Git](#).

Validation des modèles

Validation de la syntaxe

Vous pouvez vérifier la syntaxe JSON ou YAML de votre modèle à l'aide de la commande CLI [validate-template](#) ou en spécifiant votre modèle sur la console. La console effectue automatiquement la validation. Pour de plus amples informations, veuillez consulter [Création d'une pile depuis la CloudFormation console](#).

Cependant, ces méthodes vérifient uniquement la syntaxe de votre modèle et ne valident pas les valeurs de propriété que vous avez spécifiées pour une ressource.

Outils de validation supplémentaires

Pour des validations plus complexes et des vérifications de la bonne pratique, vous pouvez utiliser des outils supplémentaires tels que :

- [CloudFormation Linter \(cfn-lint\)](#) — Validez les modèles par rapport aux schémas du [CloudFormation fournisseur de ressources](#). Comprend la vérification des valeurs valides pour les propriétés des ressources et les bonnes pratiques.
- [CloudFormation Rain \(rain fmt\)](#) — Formatez vos CloudFormation modèles selon une norme cohérente ou reformatez un modèle de JSON en YAML (ou de YAML en JSON). Il conserve les commentaires lors de l'utilisation de YAML et passe à la syntaxe courte lorsque cela est possible pour l'utilisation des fonctions intrinsèques.

Premiers pas avec les modèles

Pour commencer à créer un CloudFormation modèle, procédez comme suit :

1. Choisissez les ressources : identifiez les AWS ressources que vous souhaitez inclure dans votre pile, telles que EC2 les instances VPCs, les groupes de sécurité, etc.
2. Rédiger le modèle : rédigez le modèle au format JSON ou YAML, en définissant les ressources et leurs propriétés.
3. Enregistrer le modèle : enregistrez le modèle localement avec une extension de fichier telle que : `.json`, `.yaml` ou `.txt`.
4. Valider le modèle : validez le modèle à l'aide des méthodes décrites dans la section [Validation des modèles](#).
5. Créer une pile : créez une pile à l'aide du modèle validé.

Prévoyez d'utiliser la référence du CloudFormation modèle

Lorsque vous rédigez vos modèles, vous pouvez trouver la documentation relative à la syntaxe détaillée des différents types de ressources dans la [Référence des types de ressources et de propriétés AWS](#).

Souvent, vos modèles de pile nécessiteront des fonctions intrinsèques pour attribuer des valeurs de propriété qui ne sont pas disponibles avant l'exécution et des attributs spéciaux pour contrôler le comportement des ressources. Lorsque vous rédigez votre modèle, veuillez vous référer aux ressources suivantes pour obtenir des conseils :

- [Référence des fonctions intrinsèques](#) : voici quelques fonctions intrinsèques couramment utilisées :
 - `Ref` : récupère la valeur d'un paramètre ou l'ID physique d'une ressource.
 - `Sub` : remplace les espaces réservés dans les chaînes par des valeurs réelles.
 - `GetAtt` : renvoie la valeur d'un attribut d'une ressource dans le modèle.
 - `Join` : joint un ensemble de valeurs dans une seule chaîne.
- [Référence des attributs de ressource](#) : voici quelques attributs spéciaux couramment utilisés :
 - `DependsOn` : utilisez cet attribut pour spécifier qu'une ressource doit être créée après une autre.
 - `DeletionPolicy`— Utilisez cet attribut pour spécifier CloudFormation comment gérer la suppression d'une ressource.

Exemples de modèle

CloudFormation fournit des modèles de stack open source que vous pouvez utiliser pour démarrer. Pour plus d'informations, consultez la section [CloudFormation Exemples de modèles](#) sur le GitHub site Web.

Veuillez noter que ces modèles ne sont pas destinés à être utilisés en production. Il est recommandé de prendre le temps de comprendre leur fonctionnement, de les adapter à vos besoins et de vous assurer qu'ils répondent aux normes de conformité de votre entreprise.

Chaque modèle de ce référentiel passe les tests [CloudFormation Linter](#) (cfn-lint), ainsi qu'un ensemble de AWS CloudFormation Guard règles de base basé sur le Center for Internet Security (CIS) Top 20, à l'exception de certaines règles pour lesquelles il était logique de concentrer l'échantillon sur un seul cas d'utilisation.

CloudFormation format de modèle

Vous pouvez créer des CloudFormation modèles au format JSON ou YAML. Les deux formats ont le même objectif, mais offrent des avantages distincts en termes de lisibilité et de complexité.

- **JSON** : JSON est un format d'échange de données léger, facile à analyser et à générer par les machines. Cependant, il peut être fastidieux à lire et à écrire pour les humains, en particulier pour les configurations complexes. En JSON, le modèle est structuré à l'aide d'accolades imbriquées `{ }` et de crochets `[ ]` pour définir les ressources, les paramètres et d'autres composants. Sa syntaxe nécessite la déclaration explicite de chaque élément, ce qui peut rendre le modèle verbeux, mais garantit le strict respect d'un format structuré.
- **YAML** : YAML est conçu pour être plus lisible par l'homme et moins verbeux que JSON. Il utilise des indentations plutôt que des accolades et des crochets pour indiquer l'imbrication, ce qui peut faciliter le fait de visualiser la hiérarchie des ressources et des paramètres. YAML est souvent préféré pour sa clarté et sa facilité d'utilisation, en particulier lorsqu'il s'agit de modèles plus complexes. Cependant, le recours aux indentations dans YAML peut entraîner des erreurs si l'espacement n'est pas cohérent, ce qui nécessite une attention particulière pour maintenir la précision.

Structure du modèle

CloudFormation les modèles sont divisés en différentes sections, et chaque section est conçue pour contenir un type d'information spécifique. Certaines sections doivent être déclarées dans un ordre précis, tandis que pour d'autres, l'ordre n'a pas d'importance. Cependant, lorsque vous créez votre modèle, il peut être utile d'utiliser l'ordre logique indiqué dans les exemples suivants, car les valeurs d'une section peuvent faire référence à des valeurs d'une section précédente.

Lorsque vous créez des modèles, veuillez ne pas utiliser de sections principales en double, telles que la section `Resources`. Bien qu'il soit CloudFormation possible d'accepter le modèle, celui-ci aura un comportement indéfini lors du traitement du modèle et risque de fournir des ressources de manière incorrecte ou de renvoyer des erreurs inexplicables.

JSON

L'exemple suivant montre la structure d'un modèle au format JSON avec toutes les sections disponibles.

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  }
}
```

```
},  
  
"Parameters" : {  
    set of parameters  
},  
  
"Rules" : {  
    set of rules  
},  
  
"Mappings" : {  
    set of mappings  
},  
  
"Conditions" : {  
    set of conditions  
},  
  
"Transform" : {  
    set of transforms  
},  
  
"Resources" : {  
    set of resources  
},  
  
"Outputs" : {  
    set of outputs  
}  
}
```

YAML

L'exemple suivant montre la structure d'un modèle au format YAML avec toutes les sections disponibles.

```
---  
AWSTemplateFormatVersion: version date  
  
Description:  
    String  
  
Metadata:
```

template metadata

Parameters:

set of parameters

Rules:

set of rules

Mappings:

set of mappings

Conditions:

set of conditions

Transform:

set of transforms

Resources:

set of resources

Outputs:

set of outputs

Commentaires

Dans les modèles au format JSON, les commentaires ne sont pas pris en charge. De par sa conception, JSON n'inclut pas de syntaxe pour les commentaires, ce qui signifie que vous ne pouvez pas ajouter de commentaires directement dans la structure JSON. Toutefois, si vous devez inclure des notes explicatives ou de la documentation, vous pouvez envisager d'ajouter des métadonnées. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Dans les modèles au format YAML, vous pouvez inclure des commentaires en ligne à l'aide du symbole #.

L'exemple suivant illustre un modèle YAML avec des commentaires en ligne.

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample CloudFormation template with YAML comments.
# Resources section
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
```

```
Properties:
  # Linux AMI
  ImageId: ami-1234567890abcdef0
  InstanceType: t2.micro
  KeyName: MyKey
  BlockDeviceMappings:
    - DeviceName: /dev/sdm
      Ebs:
        VolumeType: io1
        Iops: 200
        DeleteOnTermination: false
        VolumeSize: 20
```

Spécifications

CloudFormation prend en charge les spécifications JSON et YAML suivantes :

JSON

CloudFormation suit la norme JSON ECMA-404. Pour plus d'informations sur le format JSON, consultez <http://www.json.org>.

YAML

CloudFormation prend en charge la spécification YAML version 1.1 à quelques exceptions près. CloudFormation ne prend pas en charge les fonctionnalités suivantes :

- les balises `binary`, `omap`, `pairs`, `set` et `timestamp` ;
- les alias ;
- les fusions de hachage.

Pour plus d'informations sur YAML, consultez <https://yaml.org/>.

En savoir plus

Pour chaque ressource que vous spécifiez dans votre modèle, vous définissez ses propriétés et ses valeurs à l'aide des règles syntaxiques spécifiques de JSON ou YAML. Pour plus d'informations sur la syntaxe des modèles pour chaque format, consultez [CloudFormation sections du modèle](#).

Utilisation d'expressions régulières dans les CloudFormation modèles

[Vous pouvez utiliser des expressions régulières \(communément appelées expressions régulières\) à plusieurs endroits de vos CloudFormation modèles, par exemple pour la AllowedPattern propriété lors de la création d'un paramètre de modèle.](#)

Toutes les expressions régulières CloudFormation sont conformes à la syntaxe Java regex. Pour une description complète de cette syntaxe et de ses constructions, consultez java.util.regex.Pattern.

Si vous écrivez votre CloudFormation modèle dans la syntaxe JSON, vous devez éviter toute barre oblique inverse (\) dans votre expression régulière en ajoutant une barre oblique inverse supplémentaire. Cela s'explique par le fait que JSON interprète la barre oblique inversée comme un caractère d'échappement, et que vous devez l'échapper pour qu'elle soit traitée comme un caractère littéral dans l'expression régulière.

Par exemple, si vous incluez un \d dans l'expression régulière afin de désigner un caractère numérique, vous devrez écrire \\d dans votre modèle JSON.

Dans l'exemple suivant, la propriété AllowedPattern spécifie une expression régulière correspondant à quatre chiffres (\d{4}). Cependant, étant donné que l'expression régulière est définie dans un modèle JSON, le caractère de barre oblique inversée doit être échappé par une barre oblique inversée (\\d) supplémentaire.

```
{
  "Parameters": {
    "MyParameter": {
      "Type": "String",
      "AllowedPattern": "\\d{4}"
    }
  }
}
```

Si vous écrivez votre CloudFormation modèle dans la syntaxe YAML, vous devez entourer l'expression régulière de guillemets simples («»). Aucun échappement supplémentaire n'est nécessaire.

```
Parameters:
  MyParameter:
    Type: String
```

```
AllowedPattern: '\\d{4}'
```

Note

Les expressions régulières dans ne CloudFormation sont prises en charge qu'à des fins de validation dans des contextes spécifiques tels que `AllowedPattern`. Elles ne sont pas prises en charge en tant qu'opérations de correspondance de modèles dans les fonctions CloudFormation intrinsèques `Fn::Equals`, telles que celles qui effectuent uniquement une comparaison de chaînes exacte, et non une correspondance de modèles.

CloudFormation sections du modèle

Chaque CloudFormation modèle comprend une ou plusieurs sections, chacune ayant un objectif spécifique.

La section `Ressources` est obligatoire dans chaque CloudFormation modèle et constitue le cœur du modèle. Cette section décrit les ressources de la pile et leurs propriétés, telles qu'une EC2 instance Amazon ou un compartiment Amazon S3. Chaque ressource est définie avec un identifiant logique unique, un type et des détails de configuration spécifiques.

La section `Parameters`, bien que facultative, joue un rôle important dans la flexibilité des modèles. Elle permet aux utilisateurs de transmettre des valeurs au moment de l'exécution lors de la création ou de la mise à jour d'une pile. Ces paramètres peuvent être référencés dans les sections `Resources` et `Outputs`, ce qui permet une personnalisation sans modifier le modèle lui-même. Par exemple, vous pouvez utiliser des paramètres pour spécifier des types d'instances ou des paramètres d'environnement qui varient d'un déploiement à l'autre.

La section `Outputs`, également facultative, définit les valeurs qui sont renvoyées lors de l'affichage des propriétés d'une pile. Les résultats fournissent des informations utiles, telles que des identifiants de ressources URLs, qui peuvent être utilisées à des fins opérationnelles ou pour être intégrées à d'autres piles. Cette section aide les utilisateurs à récupérer et à utiliser des détails importants sur les ressources créées par le modèle.

D'autres sections facultatives incluent `Mappings`, qui fonctionne comme des tables de recherche pour gérer les valeurs conditionnelles. Avec les mappages, vous définissez des paires clé-valeur et les utilisez avec la fonction intrinsèque `Fn::FindInMap` dans les sections `Resources` et `Outputs`. Cela est utile pour les scénarios dans lesquels vous devez ajuster les configurations en fonction de conditions telles que Région AWS l'environnement.

Les sections Metadata et Rules, bien que moins couramment utilisées, offrent des fonctionnalités supplémentaires. Metadata peut inclure des informations supplémentaires sur le modèle, tandis que Rules valide un paramètre ou une combinaison de paramètres lors de la création ou de la mise à jour d'une pile, garantissant ainsi qu'ils répondent à des critères spécifiques. La section Conditions améliore encore la flexibilité en contrôlant si certaines ressources sont créées ou si des propriétés se voient attribuer une valeur en fonction de conditions telles que le type d'environnement.

Enfin, la section Transform est utilisée pour appliquer des macros pendant le traitement du modèle. Pour les applications sans serveur (également appelées applications Lambda), il indique la version du [modèle d'AWS application sans serveur AWS \(SAM\)](#) à utiliser. Lorsque vous spécifiez une transformation, vous pouvez utiliser AWS SAM la syntaxe pour déclarer les ressources dans votre modèle. Le modèle définit la syntaxe que vous pouvez utiliser et comment elle est traitée. Vous pouvez également utiliser la `AWS::Include` transformation pour inclure des extraits de modèle stockés séparément du modèle principal CloudFormation .

Les rubriques suivantes fournissent plus d'informations et des exemples d'utilisation de chaque section.

Rubriques

- [CloudFormation Ressourcessyntaxe du modèle](#)
- [CloudFormation Parameterssyntaxe du modèle](#)
- [CloudFormation Outputsyntaxe du modèle](#)
- [CloudFormation Mappingssyntaxe du modèle](#)
- [CloudFormation Metadatasyntaxe du modèle](#)
- [CloudFormation Rulesyntaxe du modèle](#)
- [CloudFormation Conditionssyntaxe du modèle](#)
- [CloudFormation Transformsection des modèles](#)
- [CloudFormation syntaxe de version du format de modèle](#)
- [CloudFormation Descriptionsyntaxe du modèle](#)

CloudFormation Ressourcessyntaxe du modèle

La Resources section est une section de niveau supérieur obligatoire dans un CloudFormation modèle. Il déclare les AWS ressources que vous souhaitez CloudFormation provisionner et configurer dans le cadre de votre stack.

Syntaxe

La section Resources utilise la syntaxe suivante :

JSON

```
"Resources" : {  
  "LogicalResourceName1" : {  
    "Type" : "AWS::ServiceName::ResourceType",  
    "Properties" : {  
      "PropertyName1" : "PropertyValue1",  
      ...  
    }  
  },  
  
  "LogicalResourceName2" : {  
    "Type" : "AWS::ServiceName::ResourceType",  
    "Properties" : {  
      "PropertyName1" : "PropertyValue1",  
      ...  
    }  
  }  
}
```

YAML

```
Resources:  
  LogicalResourceName1:  
    Type: AWS::ServiceName::ResourceType  
    Properties:  
      PropertyName1: PropertyValue1  
      ...  
  
  LogicalResourceName2:  
    Type: AWS::ServiceName::ResourceType  
    Properties:  
      PropertyName1: PropertyValue1  
      ...
```

ID logique (également appelé nom logique)

Dans un CloudFormation modèle, les ressources sont identifiées par leurs noms de ressources logiques. Ces noms doivent être alphanumériques (A-Za-z0-9) et uniques dans le modèle. Les noms logiques sont utilisés pour référencer les ressources d'autres sections du modèle.

Type de ressource

Chaque ressource doit avoir un attribut `Type`, qui définit le type de ressource AWS dont il s'agit. L'attribut `Type` a le format `AWS::ServiceName::ResourceType`. Par exemple, l'attribut `Type` pour un compartiment Amazon S3 est `AWS::S3::Bucket`.

Pour obtenir la liste complète des types de ressources pris en charge, consultez la [Référence des types de ressources et de propriétés AWS](#).

Propriétés de ressource

Les propriétés des ressources sont des options supplémentaires que vous pouvez spécifier pour définir les détails de configuration d'un type de ressource spécifique. Certaines propriétés sont obligatoires, tandis que d'autres sont facultatives. Certaines propriétés ont des valeurs par défaut, leur spécification est donc facultative.

Pour plus d'informations sur les propriétés prises en charge pour chaque type de ressource, consultez les rubriques de la [Référence des types de ressources et de propriétés AWS](#).

Les valeurs de propriété peuvent être des chaînes littérales, des listes de chaînes, des booléens, des références dynamiques, des références de paramètres, des pseudo-références ou la valeur renvoyée par une fonction. Les exemples suivants vous montrent comment déclarer différents types de valeurs de propriété :

JSON

```
"Properties" : {  
  "String" : "A string value",  
  "Number" : 123,  
  "LiteralList" : [ "first-value", "second-value" ],  
  "Boolean" : true  
}
```

YAML

```
Properties:
```

```
String: A string value
Number: 123
LiteralList:
  - first-value
  - second-value
Boolean: true
```

ID physique

En plus de l'ID logique, certaines ressources ont également un ID physique, qui constitue le nom réel affecté à cette ressource (par exemple, un ID d'instance EC2 ou un nom de compartiment S3). Utilisez les identifiants physiques pour identifier les ressources en dehors des CloudFormation modèles, mais uniquement après leur création. Par exemple, supposons que vous attribuez l'ID logique `MyEC2Instance` à une ressource d'instance EC2. Lors de la CloudFormation création de l'instance, il génère et attribue CloudFormation automatiquement un identifiant physique (tel que `i-1234567890abcdef0`) à l'instance. Vous pouvez utiliser cet ID physique pour identifier l'instance et afficher ses propriétés (telles que le nom DNS) via la console Amazon EC2.

Pour les compartiments Amazon S3 et de nombreuses autres ressources, génère CloudFormation automatiquement un nom physique unique pour la ressource si vous ne le spécifiez pas explicitement. Ce nom physique est basé sur une combinaison du nom de la CloudFormation pile, du nom logique de la ressource spécifié dans le CloudFormation modèle et d'un identifiant unique. Par exemple, si vous avez un compartiment Amazon S3 dont le nom logique figure `MyBucket` dans une pile nommée `MyStack`, vous CloudFormation pouvez le nommer avec le nom physique suivant `MyStack-MyBucket-abcdefghijkl1`.

Pour les ressources qui prennent en charge les noms personnalisés, vous pouvez attribuer vos propres noms physiques afin de vous aider à identifier rapidement les ressources. Par exemple, vous pouvez attribuer le nom `MyPerformanceLogs` à un compartiment S3 qui stocke des journaux. Pour plus d'informations, consultez [Type de nom](#).

Référencement des ressources

Souvent, vous devez définir les propriétés d'une ressource en fonction du nom ou de la propriété d'une autre ressource. Par exemple, vous pouvez créer une instance EC2 qui utilise des groupes de sécurité EC2 ou une CloudFront distribution soutenue par un compartiment S3. Toutes ces ressources peuvent être créées dans le même CloudFormation modèle.

CloudFormation fournit des fonctions intrinsèques que vous pouvez utiliser pour faire référence à d'autres ressources et à leurs propriétés. Ces fonctions vous permettent de créer des dépendances entre les ressources et de transmettre des valeurs d'une ressource à une autre.

La fonction **Ref**

La Ref fonction est couramment utilisée pour récupérer une propriété d'identification des ressources définies dans le même CloudFormation modèle. Ce qu'elle renvoie dépend du type de ressource. Pour la plupart des ressources, elle renvoie le nom physique de la ressource. Cependant, pour certains types de ressources, elle peut renvoyer une valeur différente, telle qu'une adresse IP pour une ressource `AWS::EC2::EIP` ou un Amazon Resource Name (ARN) pour une rubrique Amazon SNS.

Les exemples suivants illustrent comment utiliser la fonction Ref dans les propriétés. Dans chacun de ces exemples, la fonction Ref renvoie le nom réel de la ressource `LogicalResourceName` déclarée ailleurs dans le modèle. L'exemple de syntaxe `!Ref` dans l'exemple YAML est simplement une façon plus concise d'écrire la fonction Ref.

JSON

```
"Properties" : {  
  "PropertyName" : { "Ref" : "LogicalResourceName" }  
}
```

YAML

```
Properties:  
  PropertyName1:  
    Ref: LogicalResourceName  
  PropertyName2: !Ref LogicalResourceName
```

Pour plus d'informations sur la fonction Ref, consultez [Ref](#).

La fonction **Fn::GetAtt**

La fonction Ref est utile si le paramètre ou la valeur renvoyée pour une ressource correspond exactement à ce que vous voulez. Cependant, vous pouvez avoir besoin d'autres attributs d'une ressource. Par exemple, si vous souhaitez créer une CloudFront distribution avec une origine S3, vous devez spécifier l'emplacement du compartiment à l'aide d'une adresse de type DNS. Plusieurs ressources possèdent des attributs supplémentaires dont vous pouvez utiliser les valeurs dans le modèle. Pour obtenir ces attributs, utilisez la fonction `Fn::GetAtt`.

Les exemples suivants illustrent comment utiliser la fonction `GetAtt` dans les propriétés. La fonction `Fn::GetAtt` prend deux paramètres : le nom logique de la ressource et le nom de l'attribut à récupérer. L'exemple de syntaxe `!GetAtt` dans l'exemple YAML est simplement une façon plus concise d'écrire la fonction `GetAtt`.

JSON

```
"Properties" : {  
  "PropertyName" : {  
    "Fn::GetAtt" : [ "LogicalResourceName", "AttributeName" ]  
  }  
}
```

YAML

```
Properties:  
  PropertyName1:  
    Fn::GetAtt:  
      - LogicalResourceName  
      - AttributeName  
  PropertyName2: !GetAtt LogicalResourceName.AttributeName
```

Pour plus d'informations sur la fonction `GetAtt`, consultez [Fn::GetAtt](#).

Exemples

Les exemples suivants montrent comment déclarer des ressources et comment les CloudFormation modèles peuvent faire référence à d'autres ressources définies dans le même modèle et dans AWS des ressources existantes.

Rubriques

- [Déclaration d'une seule ressource avec un nom personnalisé](#)
- [Référencement d'autres ressources avec la fonction Ref](#)
- [Référencement des attributs de ressource à l'aide de la fonction Fn::GetAtt](#)

Déclaration d'une seule ressource avec un nom personnalisé

Les exemples suivants déclarent une seule ressource de type `AWS::S3::Bucket` avec le nom logique `MyBucket`. La propriété `BucketName` est définie sur `amzn-s3-demo-bucket`, qui doit être remplacé par le nom souhaité pour votre compartiment S3.

Si vous utilisez cette déclaration de ressource pour créer une pile, vous CloudFormation créez un compartiment Amazon S3 avec les paramètres par défaut. Pour d'autres ressources, telles qu'une instance Amazon EC2 ou un groupe Auto Scaling, des informations supplémentaires sont CloudFormation nécessaires.

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "BucketName": "amzn-s3-demo-bucket"
      }
    }
  }
}
```

YAML

```
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
```

Référencement d'autres ressources avec la fonction **Ref**

Les exemples suivants montrent une déclaration de ressource qui définit une instance EC2 et un groupe de sécurité. La ressource Ec2Instance référence la ressource InstanceSecurityGroup dans le cadre de sa propriété SecurityGroupIds à l'aide de la fonction Ref. Elle inclut également un groupe de sécurité existant (sg-12a4c434) qui n'est pas déclaré dans le modèle. Les chaînes littérales permettent de faire référence à des ressources AWS existantes.

JSON

```
{
  "Resources": {
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
```

```

        "SecurityGroupIds": [
            {
                "Ref": "InstanceSecurityGroup"
            },
            "sg-12a4c434"
        ],
        "KeyName": "MyKey",
        "ImageId": "ami-1234567890abcdef0"
    },
    "InstanceSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable SSH access via port 22",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "CidrIp": "0.0.0.0/0"
                }
            ]
        }
    }
}

```

YAML

```

Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
        - sg-12a4c434
      KeyName: MyKey
      ImageId: ami-1234567890abcdef0
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:

```



```
- IpProtocol: tcp
  FromPort: 22
  ToPort: 22
  CidrIp: 0.0.0.0/0
```

Référencement des attributs de ressource à l'aide de la fonction **Fn::GetAtt**

Les exemples suivants présentent une déclaration de ressource qui définit une ressource CloudFront de distribution et un compartiment S3. La ressource `MyDistribution` spécifie le nom DNS de la ressource `MyBucket` à l'aide de la fonction `Fn::GetAtt` pour obtenir l'attribut `DomainName` du compartiment. Vous remarquerez que la fonction `Fn::GetAtt` répertorie ses deux paramètres dans un tableau. Pour les fonctions qui acceptent plusieurs paramètres, un tableau permet de spécifier ces derniers.

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket"
    },
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "Origins": [
            {
              "DomainName": {
                "Fn::GetAtt": [
                  "MyBucket",
                  "DomainName"
                ]
              },
              "Id": "MyS3Origin",
              "S3OriginConfig": {}
            }
          ],
          "Enabled": "true",
          "DefaultCacheBehavior": {
            "TargetOriginId": "MyS3Origin",
            "ForwardedValues": {
              "QueryString": "false"
            }
          }
        }
      }
    }
  }
}
```

```

        "ViewerProtocolPolicy": "allow-all"
    }
}
}
}
}
}
}

```

YAML

```

Resources:
  MyBucket:
    Type: AWS::S3::Bucket
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: !GetAtt
            - MyBucket
            - DomainName
            Id: MyS3Origin
            S3OriginConfig: {}
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: MyS3Origin
          ForwardedValues:
            QueryString: 'false'
          ViewerProtocolPolicy: allow-all

```

CloudFormation Parameters syntaxe du modèle

Utilisez la section **Parameters** facultative pour personnaliser vos modèles. Les paramètres vous permettent de saisir des valeurs personnalisées dans votre modèle chaque fois que vous créez ou mettez à jour une pile. En utilisant des paramètres dans vos modèles, vous pouvez créer des modèles réutilisables et flexibles qui peuvent être adaptés à des scénarios spécifiques.

En définissant les paramètres du type approprié, vous pouvez choisir parmi une liste d'identificateurs de ressources existantes lorsque vous utilisez la console pour créer votre pile. Pour de plus amples informations, veuillez consulter [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#).

Les paramètres sont un moyen courant de spécifier les valeurs des propriétés des ressources de pile. Cependant, certains paramètres peuvent dépendre de la région ou être quelque peu complexes à comprendre pour les utilisateurs en raison d'autres conditions ou dépendances. Dans ces cas, vous pouvez intégrer une logique dans le modèle lui-même afin que les utilisateurs puissent spécifier des valeurs plus simples (ou aucune valeur) pour obtenir les résultats souhaités, par exemple en utilisant un mappage. Pour de plus amples informations, veuillez consulter [CloudFormation Mappingssyntaxe du modèle](#).

Syntaxe

Vous déclarez les paramètres dans la section `Parameters` d'un modèle, qui utilise la syntaxe générale suivante :

JSON

```
"Parameters" : {  
  "ParameterLogicalID" : {  
    "Description": "Information about the parameter",  
    "Type" : "DataType",  
    "Default" : "value",  
    "AllowedValues" : ["value1", "value2"]  
  }  
}
```

YAML

```
Parameters:  
  ParameterLogicalID:  
    Description: Information about the parameter  
    Type: DataType  
    Default: value  
    AllowedValues:  
      - value1  
      - value2
```

Un paramètre contient une liste d'attributs qui définissent sa valeur, ainsi que les contraintes de cette valeur. Le seul attribut requis est `Type`, qui peut être `String`, `Number`, ou un type de paramètre CloudFormation fourni par -. Vous pouvez également ajouter un attribut `Description` qui décrit le type de valeur que vous devez spécifier. Le nom et la description du paramètre apparaissent dans la page Spécifier les paramètres lorsque vous utilisez le modèle dans l'assistant Créer une pile.

 Note

Par défaut, la CloudFormation console répertorie les paramètres d'entrée par ordre alphabétique selon leur identifiant logique. Pour annuler cet ordre par défaut et regrouper les paramètres associés, vous pouvez utiliser la clé de métadonnées `AWS::CloudFormation::Interface` de votre modèle. Pour de plus amples informations, veuillez consulter [Organisation des CloudFormation paramètres à l'aide de AWS::CloudFormation::Interface métadonnées](#).

Pour les paramètres avec des valeurs par défaut, CloudFormation utilise les valeurs par défaut sauf si les utilisateurs spécifient une autre valeur. Si vous omettez l'attribut par défaut, les utilisateurs doivent spécifier une valeur pour ce paramètre. Cependant, vous ne pouvez pas garantir que cette valeur sera correcte. Pour valider la valeur d'un paramètre, vous pouvez déclarer des contraintes ou spécifier un type de paramètre AWS spécifique.

Pour les paramètres sans valeur par défaut, les utilisateurs doivent spécifier une valeur de nom de clé lors de la création de la pile. S'ils ne le font pas, ils CloudFormation ne parviennent pas à créer la pile et génèrent une exception :

```
Parameters: [KeyName] must have values
```

Propriétés

AllowedPattern

Une expression régulière qui représente les modèles que vous voulez autoriser pour `String` ou les types `CommaDelimitedList`. Lorsqu'il est appliqué à un paramètre de type `String`, le modèle doit correspondre à la valeur entière du paramètre fournie. Lorsqu'il est appliqué à un paramètre de type `CommaDelimitedList`, le modèle doit correspondre à chaque valeur de la liste.

Obligatoire : non

AllowedValues

Tableau contenant la liste des valeurs autorisées pour le paramètre. Lorsqu'elle est appliquée à un paramètre de type `String`, la valeur du paramètre doit être l'une des valeurs autorisées. Lorsqu'elle est appliquée à un paramètre de type `CommaDelimitedList`, chaque valeur de la liste doit être l'une des valeurs autorisées spécifiées.

Obligatoire : non

 Note

Si vous utilisez le format YAML et que vous souhaitez utiliser des chaînes Yes et No pour `AllowedValues`, utilisez des guillemets simples pour empêcher l'analyseur YAML de prendre en compte ces valeurs booléennes.

ConstraintDescription

Chaîne qui explique la contrainte lorsque celle-ci n'est pas respectée. Par exemple, sans description de contrainte, un paramètre dont le modèle `[A-Za-z0-9]+` est autorisé affiche le message d'erreur suivant lorsque l'utilisateur spécifie une valeur non valide :

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

En ajoutant une description de contrainte, comme doit uniquement contenir des lettres (minuscules et majuscules) et des chiffres, vous pouvez afficher le message d'erreur personnalisé suivant :

```
Malformed input-Parameter MyParameter must only contain uppercase and lowercase letters and numbers
```

Obligatoire : non

Default

Valeur correspondant au type approprié de modèle à utiliser si aucune valeur n'est spécifiée lors de la création d'une pile. Si vous avez défini des contraintes pour le paramètre, vous devez spécifier une valeur conforme à ces contraintes.

Obligatoire : non

Description

Chaîne de maximum 4000 caractères qui décrit le paramètre.

Obligatoire : non

MaxLength

Nombre entier qui détermine le nombre maximal de caractères que vous voulez autoriser pour les types `String`.

Obligatoire : non

MaxValue

Valeur numérique qui détermine la valeur maximale que vous voulez autoriser pour les types `Number`.

Obligatoire : non

MinLength

Nombre entier qui détermine le nombre minimal de caractères que vous voulez autoriser pour les types `String`.

Obligatoire : non

MinValue

Valeur numérique qui détermine la valeur minimale que vous voulez autoriser pour les types `Number`.

Obligatoire : non

NoEcho

Pour masquer la valeur du paramètre et éviter qu'elle ne s'affiche dans la console, les outils de ligne de commande ou l'API. Si vous définissez l'attribut `NoEcho` sur `true`, CloudFormation renvoie la valeur du paramètre masquée sous forme d'astérisques (*****) pour tous les appels décrivant la pile ou les événements de la pile, à l'exception des informations stockées aux emplacements spécifiés ci-dessous.

Obligatoire : non

Important

L'utilisation de l'attribut `NoEcho` ne masque aucune information stockée dans les lieux suivants :

- La section des `Metadata` modèles. CloudFormation ne transforme, ne modifie ni n'expédie aucune information que vous incluez dans `Metadata` cette section. Pour de plus amples informations, veuillez consulter [Métadonnées](#).
- La section de modèle `Outputs` Pour de plus amples informations, veuillez consulter [Sorties](#)

- L'attribut Metadata d'une définition de ressource. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Nous vous recommandons vivement de ne pas utiliser ces mécanismes pour inclure des informations sensibles, telles que des mots de passe ou des secrets.

Important

Plutôt que d'intégrer des informations sensibles directement dans vos CloudFormation modèles, nous vous recommandons d'utiliser des paramètres dynamiques dans le modèle de pile pour référencer les informations sensibles stockées et gérées à l'extérieur CloudFormation, par exemple dans le AWS Systems Manager Parameter Store ou AWS Secrets Manager.

Pour de plus amples informations, veuillez consulter la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Important

Nous vous déconseillons vivement d'inclure des paramètres NoEcho, ou toute donnée sensible, dans les propriétés des ressources qui font partie de l'identifiant principal d'une ressource.

Lorsqu'un NoEcho paramètre est inclus dans une propriété qui constitue un identifiant de ressource principal, il CloudFormation peut utiliser la valeur réelle en texte brut dans l'identifiant de ressource principal. Cet ID de ressource peut apparaître dans toutes les sorties ou destinations dérivées.

Pour déterminer quelles propriétés de ressource constituent l'identifiant principal d'un type de ressource, reportez-vous à la documentation de référence de cette ressource dans la référence des [types de ressources et de propriétés AWS](#). Dans la section Valeurs renvoyées, la valeur de retour de la fonction Ref représente les propriétés de la ressource qui constituent l'identifiant principal du type de ressource.

Type

Type de données du paramètre (DataType).

Obligatoire : oui

CloudFormation prend en charge les types de paramètres suivants :

String

Chaîne littérale. Vous pouvez utiliser les attributs suivants pour déclarer des contraintes : `MinLength`, `MaxLength`, `Default`, `AllowedValues`, et `AllowedPattern`.

Par exemple, les utilisateurs peuvent spécifier `"MyUserName"`.

Number

Nombre entier ou valeur flottante. CloudFormation valide la valeur du paramètre sous forme de nombre ; toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une chaîne.

Vous pouvez utiliser les attributs suivants pour déclarer des contraintes : `MinValue`, `MaxValue`, `Default`, et `AllowedValues`.

Par exemple, les utilisateurs peuvent spécifier `"8888"`.

List<Number>

Tableau de nombres entiers ou de valeurs float, séparés par des virgules. CloudFormation valide la valeur du paramètre en tant que nombre. Toutefois, lorsque vous utilisez le paramètre ailleurs dans le modèle (par exemple, avec la fonction intrinsèque `Ref`), la valeur du paramètre devient une liste de chaînes.

Les utilisateurs peuvent par exemple préciser `"80, 20"` et la fonction `Ref` générera `["80", "20"]`.

CommaDelimitedList

Tableau de chaînes littérales séparées par des virgules. Le nombre total de chaînes doit correspondre au nombre total de virgules, plus un. En outre, les espaces de chaque chaîne de membre sont coupés.

Les utilisateurs peuvent par exemple préciser `"test, dev, prod"` et la fonction `Ref` générera `["test", "dev", "prod"]`.

Types de paramètres spécifiques àAWS

AWS des valeurs telles que les noms de paires de clés Amazon EC2 et VPC. IDs Pour de plus amples informations, veuillez consulter [Spécification de ressources existantes au moment de l'exécution](#).

Types de paramètres Systems Manager

Paramètres correspondant aux paramètres existants dans Systems Manager Parameter Store. Vous spécifiez une clé de paramètre Systems Manager comme valeur du type de paramètre Systems Manager et vous CloudFormation récupérez la dernière valeur du Parameter Store à utiliser pour la pile. Pour de plus amples informations, veuillez consulter [Spécification de ressources existantes au moment de l'exécution](#).

Consignes générales pour les paramètres

Les critères suivants s'appliquent lors de l'utilisation des paramètres :

- Vous pouvez avoir un maximum de 200 paramètres dans un CloudFormation modèle.
- Chaque paramètre doit recevoir un nom logique (également appelé ID logique) qui doit être alphanumérique et unique parmi tous les noms logiques du modèle.
- Chaque paramètre doit être associé à un type de paramètre pris en charge par CloudFormation. Pour plus d'informations, consultez [Type](#).
- Une valeur doit être affectée à chaque paramètre au moment de l'exécution CloudFormation pour approvisionner correctement la pile. Vous pouvez éventuellement spécifier une valeur par défaut CloudFormation à utiliser à moins qu'une autre valeur ne soit fournie.
- Les paramètres doivent être déclarés et référencés à partir du même modèle. Vous pouvez référencer des paramètres dans les sections Resources et Outputs du modèle.

Exemples

Rubriques

- [Un paramètre de chaîne simple](#)
- [Paramètre mot de passe](#)
- [Paramètres de référencement](#)
- [Paramètre de liste délimité par des virgules](#)
- [Renvoi d'une valeur de paramètre de liste séparée par des virgules](#)

Un paramètre de chaîne simple

L'exemple suivant déclare un paramètre nommé de type `InstanceTypeParameter` et `String`. Ce paramètre vous permet de spécifier le type d'instance Amazon EC2 pour la pile. Si aucune valeur n'est fournie lors de la création ou de la mise à jour de la pile, CloudFormation utilise la valeur par défaut `t2.micro`.

JSON

```
"Parameters" : {  
  "InstanceTypeParameter" : {  
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro.",  
    "Type" : "String",  
    "Default" : "t2.micro",  
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"]  
  }  
}
```

YAML

```
Parameters:  
  InstanceTypeParameter:  
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.  
    Type: String  
    Default: t2.micro  
    AllowedValues:  
      - t2.micro  
      - m1.small  
      - m1.large
```

Paramètre mot de passe

L'exemple suivant déclare un paramètre nommé `DBPwd` de type `String` sans valeur par défaut. La propriété `NoEcho` est définie sur `true` pour empêcher l'affichage de la valeur du paramètre dans les descriptions de pile. La longueur minimale qui peut être spécifiée s'élève à 1, et la longueur maximale à 41. Le modèle accepte les chiffres et les lettres minuscules et majuscules. Cet exemple illustre également l'utilisation d'une expression régulière pour la propriété `AllowedPattern`.

JSON

```
"Parameters" : {  
  "DBPwd" : {
```

```
"NoEcho" : "true",
"Description" : "The database admin account password",
"Type" : "String",
"MinLength" : "1",
"MaxLength" : "41",
"AllowedPattern" : "^[a-zA-Z0-9]*$"
}
```

YAML

```
Parameters:
  DBPwd:
    NoEcho: true
    Description: The database admin account password
    Type: String
    MinLength: 1
    MaxLength: 41
    AllowedPattern: ^[a-zA-Z0-9]*$
```

Paramètres de référencement

Vous utilisez la fonction intrinsèque `Ref` pour référencer un paramètre et CloudFormation utilise la valeur du paramètre pour allouer la pile. Vous pouvez faire référence à des paramètres dans les sections `Resources` et `Outputs` du même modèle.

Dans l'exemple suivant, la propriété `InstanceType` de l'instance EC2 ressource fait référence à la valeur du paramètre `InstanceTypeParameter` :

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "InstanceType" : { "Ref" : "InstanceTypeParameter" },
    "ImageId" : "ami-0ff8a91507f77f867"
  }
}
```

YAML

```
Ec2Instance:
```

```
Type: AWS::EC2::Instance
Properties:
  InstanceType:
    Ref: InstanceTypeParameter
  ImageId: ami-0ff8a91507f77f867
```

Paramètre de liste délimité par des virgules

Le type de paramètre `CommaDelimitedList` peut être utile lorsque vous devez fournir plusieurs valeurs pour une seule propriété. L'exemple suivant déclare un paramètre nommé `DbSubnetIpBlocks` avec une valeur par défaut de trois blocs CIDR séparés par des virgules.

JSON

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three CIDR blocks",
    "Type": "CommaDelimitedList",
    "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

YAML

```
Parameters:
  DbSubnetIpBlocks:
    Description: "Comma-delimited list of three CIDR blocks"
    Type: CommaDelimitedList
    Default: "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
```

Renvoi d'une valeur de paramètre de liste séparée par des virgules

Pour faire référence à une valeur spécifique dans une liste de paramètres séparés par des virgules, utilisez la fonction intrinsèque `Fn::Select` dans la section `Resources` de votre modèle. Transmettez la valeur d'index de l'objet souhaité et une liste d'objets, comme indiqué dans l'exemple suivant.

JSON

```
{
  "Parameters": {
    "VPC": {
```

```

        "Type": "String",
        "Default": "vpc-123456"
    },
    "VpcAzs": {
        "Type": "CommaDelimitedList",
        "Default": "us-west-2a, us-west-2b, us-west-2c"
    },
    "DbSubnetIpBlocks": {
        "Type": "CommaDelimitedList",
        "Default": "172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26"
    }
},
"Resources": {
    "DbSubnet1": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "AvailabilityZone": {
                "Fn::Select": [
                    0,
                    {
                        "Ref": "VpcAzs"
                    }
                ]
            },
            "VpcId": {
                "Ref": "VPC"
            },
            "CidrBlock": {
                "Fn::Select": [
                    0,
                    { "Ref": "DbSubnetIpBlocks" }
                ]
            }
        }
    },
    "DbSubnet2": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "AvailabilityZone": {
                "Fn::Sub": [
                    "${AWS::Region}${AZ}",
                    {
                        "AZ": {
                            "Fn::Select": [

```

```

        1,
        { "Ref": "VpcAzs" }
      ]
    }
  ],
  "VpcId": {
    "Ref": "VPC"
  },
  "CidrBlock": {
    "Fn::Select": [
      1,
      { "Ref": "DbSubnetIpBlocks" }
    ]
  }
},
"DbSubnet3": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "AvailabilityZone": {
      "Fn::Sub": [
        "${AWS::Region}${AZ}",
        {
          "AZ": {
            "Fn::Select": [
              2,
              { "Ref": "VpcAzs" }
            ]
          }
        }
      ]
    },
    "VpcId": {
      "Ref": "VPC"
    },
    "CidrBlock": {
      "Fn::Select": [
        2,
        { "Ref": "DbSubnetIpBlocks" }
      ]
    }
  }
}
}

```

```

    }
  }
}
```

YAML

```

Parameters:
  VPC:
    Type: String
    Default: vpc-123456
  VpcAzs:
    Type: CommaDelimitedList
    Default: us-west-2a, us-west-2b, us-west-2c
  DbSubnetIpBlocks:
    Type: CommaDelimitedList
    Default: 172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26
Resources:
  DbSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Select
        - 0
        - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 0
        - !Ref DbSubnetIpBlocks
  DbSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
            - 1
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 1
        - !Ref DbSubnetIpBlocks
  DbSubnet3:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
```

```
- ${AWS::Region}${AZ}
- AZ: !Select
  - 2
  - !Ref VpcAzs
VpcId: !Ref VPC
CidrBlock: !Select
  - 2
  - !Ref DbSubnetIpBlocks
```

Ressources connexes

CloudFormation prend également en charge l'utilisation de références dynamiques pour spécifier les valeurs des propriétés de manière dynamique. Par exemple, vous pouvez avoir besoin de référencer des chaînes sécurisées stockées dans le magasin de paramètres Systems Manager. Pour de plus amples informations, veuillez consulter [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

Vous pouvez également utiliser des pseudo-paramètres dans une fonction Ref ou Sub une variable pour remplir dynamiquement des valeurs. Pour de plus amples informations, veuillez consulter [Obtenir des AWS valeurs à l'aide de pseudo-paramètres](#).

CloudFormation Outputsyntaxe du modèle

La section `Outputs` facultative déclare les valeurs de sortie pour la pile. Ces valeurs de sortie peuvent être utilisées de différentes manières :

- Capturez des informations importantes sur vos ressources : une sortie est un moyen pratique de capturer des informations importantes sur vos ressources. Par exemple, vous pouvez générer le nom de compartiment S3 d'une pile pour faciliter la recherche du compartiment. Vous pouvez afficher les valeurs de sortie dans l'onglet Sorties de la CloudFormation console ou à l'aide de la commande [describe-stacks](#)CLI.
- Références entre piles : vous pouvez importer des valeurs de sortie dans d'autres piles pour [créer des références entre piles](#). Cela est utile lorsque vous devez partager des ressources ou des configurations entre plusieurs piles.

⚠ Important

CloudFormation ne supprime ni ne masque les informations que vous incluez dans cette section. **Outputs** Nous vous recommandons vivement de ne pas utiliser cette section pour produire des informations sensibles, telles que des mots de passe ou des secrets. Les valeurs de sortie sont disponibles une fois l'opération de pile terminée. Les valeurs de sortie de la pile ne sont pas disponibles lorsque la pile se trouve dans l'un des [états](#) IN_PROGRESS suivants. Nous ne recommandons pas d'établir des dépendances entre un runtime de service et la valeur de sortie de la pile, car ces dernières peuvent ne pas être disponibles à tout moment.

Syntaxe

La section **Outputs** se compose du nom de clé **Outputs**. Vous pouvez déclarer un maximum de 200 sorties dans un modèle.

L'exemple suivant illustre la structure de la section **Outputs**.

JSON

Utilisez des accolades pour délimiter toutes les déclarations de sortie. Délimitez les différentes sorties avec des virgules.

```
"Outputs" : {  
  "OutputLogicalID" : {  
    "Description" : "Information about the value",  
    "Value" : "Value to return",  
    "Export" : {  
      "Name" : "Name of resource to export"  
    }  
  }  
}
```

YAML

```
Outputs:  
  OutputLogicalID:  
    Description: Information about the value  
    Value: Value to return
```

Export:

Name: *Name of resource to export*

Champs de sortie

La section Outputs peut inclure les champs suivants.

ID logique (également appelé nom logique)

Identificateur de la sortie actuelle. L'ID logique doit être alphanumérique (a-z, A-Z, 0-9) et ne doit correspondre à aucun autre ID logique dans le modèle.

Description (facultatif)

Type String qui décrit la valeur de sortie. La valeur de la déclaration de la description doit être une chaîne littérale dont la taille doit être comprise entre 0 et 1024 octets. Vous ne pouvez pas utiliser une fonction ou un paramètre pour spécifier la description.

Value (obligatoire)

Valeur de la propriété retournée par la commande [describe-stacks](#). La valeur d'une sortie peut inclure des littéraux, des références de paramètre, des pseudo-paramètres, une valeur de mappage ou des fonctions intrinsèques.

Export (facultatif)

Nom de la sortie de ressource à exporter pour une référence entre piles.

Vous pouvez utiliser des fonctions intrinsèques pour personnaliser la valeur Name d'une exportation.

Pour de plus amples informations, veuillez consulter [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#).

Pour associer une condition à une sortie, définissez la condition dans la section [Conditions](#) du modèle.

Exemples

Les exemples suivants illustrent la fonctionnement d'une sortie de pile.

Rubriques

- [Sortie de pile](#)

- [Personnalisez le nom de l'exportation en utilisant Fn::Sub](#)
- [Personnalisez le nom de l'exportation en utilisant Fn::Join](#)
- [Renvoie une URL construite à l'aide de Fn::Join](#)

Sortie de pile

Dans l'exemple suivant, la sortie nommée BackupLoadBalancerDNSName renvoie le nom DNS pour la ressource associée à l'ID logique BackupLoadBalancer uniquement lorsque la condition CreateProdResources a la valeur true. La sortie nommée InstanceID renvoie l'ID de l'instance EC2 avec l'ID logique EC2Instance.

JSON

```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ] },
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

YAML

```
Outputs:
  BackupLoadBalancerDNSName:
    Description: The DNSName of the backup load balancer
    Value: !GetAtt BackupLoadBalancer.DNSName
    Condition: CreateProdResources
  InstanceID:
    Description: The Instance ID
    Value: !Ref EC2Instance
```

Personnalisez le nom de l'exportation en utilisant **Fn::Sub**

Dans les exemples suivants, la sortie nommée StackVPC retourne l'ID d'un VPC et exporte ensuite la valeur pour un référencement entre piles avec le nom VPCID ajouté au nom de la pile.

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : {"Fn::Sub": "${AWS::StackName}-VPCID" }
    }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Sub "${AWS::StackName}-VPCID"
```

Pour plus d'informations sur la fonction `Fn::Sub`, consultez [Fn::Sub](#).

Personnalisez le nom de l'exportation en utilisant **Fn::Join**

Vous pouvez également utiliser cette fonction `Fn::Join` pour construire des valeurs basées sur des paramètres, des attributs de ressources et d'autres chaînes de caractères.

Les exemples suivants utilisent la fonction `Fn::Join` pour personnaliser le nom de l'exportation au lieu de la fonction `Fn::Sub`. La fonction `Fn::Join` d'exemple concatène le nom de la pile avec le nom `VPCID` en utilisant un deux-points comme séparateur.

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : { "Fn::Join" : [ ":", [ { "Ref" : "AWS::StackName" }, "VPCID" ] ] }
    }
  }
}
```

```
}  
}
```

YAML

```
Outputs:  
  StackVPC:  
    Description: The ID of the VPC  
    Value: !Ref MyVPC  
    Export:  
      Name: !Join [ ":", [ !Ref "AWS::StackName", VPCID ] ]
```

Pour plus d'informations sur la fonction `Fn::Join`, consultez [Fn::Join](#).

Renvoie une URL construite à l'aide de **Fn::Join**

Dans l'exemple suivant, pour un modèle qui crée un WordPress site, `InstallURL` la chaîne renvoyée par un appel de `Fn::Join` fonction qui concatène `http://`, le nom DNS de la ressource `ElasticLoadBalancer`, et `/wp-admin/install.php` La valeur de sortie ressemblerait à cela :

```
http://mywptests-elasticl-1gb5116sl8y5v-206169572.aws-region.elb.amazonaws.com/wp-admin/install.php
```

JSON

```
{  
  "Outputs": {  
    "InstallURL": {  
      "Value": {  
        "Fn::Join": [  
          "",  
          [  
            "http://",  
            {  
              "Fn::GetAtt": [  
                "ElasticLoadBalancer",  
                "DNSName"  
              ]  
            },  
            "/wp-admin/install.php"  
          ]  
        ]  
      }  
    }  
  }  
}
```

```

        ]
      ]
    },
    "Description": "Installation URL of the WordPress website"
  }
}
}

```

YAML

```

Outputs:
  InstallURL:
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt
          - ElasticLoadBalancer
          - DNSName
        - /wp-admin/install.php
    Description: Installation URL of the WordPress website

```

Pour plus d'informations sur la fonction `Fn::Join`, consultez [Fn::Join](#).

CloudFormation Mappingsyntaxe du modèle

La section `Mappings` facultative vous aide à créer des paires clé-valeur qui peuvent être utilisées pour spécifier des valeurs en fonction de certaines conditions ou dépendances.

Un cas d'utilisation courant de cette section `Mappings` consiste à définir des valeurs en fonction de l'emplacement Région AWS où la pile est déployée. Cela peut être réalisé à l'aide du pseudo-paramètre `AWS::Region`. Le `AWS::Region` pseudo-paramètre est une valeur qui CloudFormation correspond à la région où la pile est créée. Les pseudo-paramètres sont résolus CloudFormation lorsque vous créez la pile.

Pour récupérer des valeurs dans une carte, vous pouvez utiliser la fonction `Fn::FindInMap` intrinsèque dans la section `Resources` de votre modèle.

Syntaxe

La section `Mappings` utilise la syntaxe suivante :

JSON

```
"Mappings" : {  
  "MappingLogicalName" : {  
    "Key1" : {  
      "Name" : "Value1"  
    },  
    "Key2" : {  
      "Name" : "Value2"  
    },  
    "Key3" : {  
      "Name" : "Value3"  
    }  
  }  
}
```

YAML

```
Mappings:  
  MappingLogicalName:  
    Key1:  
      Name: Value1  
    Key2:  
      Name: Value2  
    Key3:  
      Name: Value3
```

- MappingLogicalName est le nom logique du mappage.
- Au sein du mappage, chaque association est une clé suivie d'un autre mappage.
- La clé doit être un mappage de paires nom-valeur et doit être unique dans le mappage.
- La paire nom-valeur est une étiquette, et l'ID la valeur à mapper. En donnant un nom aux valeurs, vous pouvez mapper plusieurs ensembles de valeurs à une clé.
- Les clés des mappages doivent être des chaînes littérales.
- Les valeurs peuvent être de type String ou List.

Note

Vous ne pouvez pas inclure des paramètres, des pseudo-paramètres ou des fonctions intrinsèques dans la section Mappings.

Si les valeurs d'un mappage ne sont pas actuellement utilisées par votre pile, vous ne pouvez pas mettre à jour le mappage seul. Vous devez inclure les modifications qui ajoutent, modifient ou suppriment des ressources.

Exemples

Rubriques

- [Mappage de base](#)
- [Mappage à plusieurs valeurs](#)
- [Renvoi d'une valeur de mappage](#)
- [Paramètre d'entrée et Fn::FindInMap](#)

Mappage de base

L'exemple suivant présente une section Mappings avec un mappage RegionToInstanceType, qui contient cinq clés qui correspondent à des paires nom-valeur contenant des valeurs de chaîne unique. Les clés sont les noms de région. Chaque paire nom-valeur est un type d'instance de la famille T disponible dans la région représentée par la clé. Les paires nom-valeur ont un nom (InstanceType dans l'exemple) et une valeur.

JSON

```
"Mappings" : {  
  "RegionToInstanceType" : {  
    "us-east-1"      : { "InstanceType" : "t2.micro" },  
    "us-west-1"      : { "InstanceType" : "t2.micro" },  
    "eu-west-1"      : { "InstanceType" : "t2.micro" },  
    "eu-north-1"     : { "InstanceType" : "t3.micro" },  
    "me-south-1"     : { "InstanceType" : "t3.micro" }  
  }  
}
```

YAML

```
Mappings:  
  RegionToInstanceType:
```



```

us-east-1:
  InstanceType: t2.micro
us-west-1:
  InstanceType: t2.micro
eu-west-1:
  InstanceType: t2.micro
eu-north-1:
  InstanceType: t3.micro
me-south-1:
  InstanceType: t3.micro

```

Mappage à plusieurs valeurs

L'exemple suivant comporte des clés de région qui sont mappées à deux ensembles de valeurs : l'un nommé MyAMI1 et l'autre MyAMI2.

Note

Les AMI IDs présentées dans ces exemples sont des espaces réservés à des fins de démonstration. Dans la mesure du possible, envisagez d'utiliser des références dynamiques aux paramètres AWS Systems Manager comme alternative à la section Mappings. Pour éviter de mettre à jour tous vos modèles avec un nouvel identifiant chaque fois que l'AMI que vous souhaitez utiliser change, utilisez un AWS Systems Manager paramètre pour récupérer le dernier ID d'AMI lors de la création ou de la mise à jour de la pile. Les dernières versions des paramètres couramment utilisés AMIs sont également disponibles en tant que paramètres publics dans Systems Manager. Pour plus d'informations, consultez [Obtenir des valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

JSON

```

"Mappings" : {
  "RegionToAMI" : {
    "us-east-1"      : { "MyAMI1" : "ami-12345678901234567", "MyAMI2" :
"ami-23456789012345678" },
    "us-west-1"     : { "MyAMI1" : "ami-34567890123456789", "MyAMI2" :
"ami-45678901234567890" },
    "eu-west-1"     : { "MyAMI1" : "ami-56789012345678901", "MyAMI2" :
"ami-67890123456789012" },
    "ap-southeast-1" : { "MyAMI1" : "ami-78901234567890123", "MyAMI2" :
"ami-89012345678901234" },

```

```
"ap-northeast-1" : { "MyAMI1" : "ami-90123456789012345", "MyAMI2" :
"ami-01234567890123456" }
}
}
```

YAML

```
Mappings:
  RegionToAMI:
    us-east-1:
      MyAMI1: ami-12345678901234567
      MyAMI2: ami-23456789012345678
    us-west-1:
      MyAMI1: ami-34567890123456789
      MyAMI2: ami-45678901234567890
    eu-west-1:
      MyAMI1: ami-56789012345678901
      MyAMI2: ami-67890123456789012
    ap-southeast-1:
      MyAMI1: ami-78901234567890123
      MyAMI2: ami-89012345678901234
    ap-northeast-1:
      MyAMI1: ami-90123456789012345
      MyAMI2: ami-01234567890123456
```

Renvoi d'une valeur de mappage

Vous pouvez utiliser la fonction `Fn::FindInMap` pour renvoyer une valeur portant un nom en fonction d'une clé spécifiée. L'exemple de modèle suivant contient une ressource Amazon EC2 dont la propriété `InstanceType` est attribuée par la fonction `FindInMap`. La `FindInMap` fonction spécifie la clé comme l' Région AWS endroit où la pile est créée (en utilisant le `AWS::Region` pseudo-paramètre) et `InstanceType` comme le nom de la valeur à mapper. Le `ImageId` utilise un paramètre `Systems Manager` pour récupérer dynamiquement la dernière AMI Amazon Linux 2. Pour plus d'informations sur les pseudo-paramètres, consultez [Obtenir des AWS valeurs à l'aide de pseudo-paramètres](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
```

```

    "RegionToInstanceType" : {
      "us-east-1"      : { "InstanceType" : "t2.micro" },
      "us-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-north-1"     : { "InstanceType" : "t3.micro" },
      "me-south-1"     : { "InstanceType" : "t3.micro" }
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap" : [ "RegionToInstanceType", { "Ref" :
"AWS::Region" }, "InstanceType" ]}
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Mappings:
  RegionToInstanceType:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      InstanceType: !FindInMap [RegionToInstanceType, !Ref 'AWS::Region', InstanceType]

```

Paramètre d'entrée et **Fn::FindInMap**

Le modèle d'exemple suivant montre comment créer une instance EC2 à l'aide de plusieurs mappages. Le modèle utilise des mappages imbriqués pour sélectionner automatiquement le type d'instance et le groupe de sécurité appropriés en fonction de la cible Région AWS et du type d'environnement (Dev ou Prod). Il utilise un paramètre Systems Manager pour récupérer dynamiquement la dernière AMI Amazon Linux 2.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [ "Dev", "Prod" ]
    }
  },
  "Mappings" : {
    "RegionAndEnvironmentToInstanceType" : {
      "us-east-1"      : { "Dev" : "t3.micro", "Prod" : "c5.large" },
      "us-west-1"      : { "Dev" : "t2.micro", "Prod" : "m5.large" }
    },
    "RegionAndEnvironmentToSecurityGroup" : {
      "us-east-1"      : { "Dev" : "sg-12345678", "Prod" : "sg-abcdef01" },
      "us-west-1"      : { "Dev" : "sg-ghijkl23", "Prod" : "sg-45678abc" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}",
        "InstanceType" : { "Fn::FindInMap": [ "RegionAndEnvironmentToInstanceType",
        { "Ref": "AWS::Region" }, { "Ref": "EnvironmentType" } ] },
        "SecurityGroupIds" : [{ "Fn::FindInMap" :
        [ "RegionAndEnvironmentToSecurityGroup", { "Ref" : "AWS::Region" }, { "Ref" :
        "EnvironmentType" } ] } ]
      }
    }
  }
}
```

```
}
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  RegionAndEnvironmentToInstanceType:
    us-east-1:
      Dev: t3.micro
      Prod: c5.large
    us-west-1:
      Dev: t2.micro
      Prod: m5.large
  RegionAndEnvironmentToSecurityGroup:
    us-east-1:
      Dev: sg-12345678
      Prod: sg-abcdef01
    us-west-1:
      Dev: sg-ghijkl23
      Prod: sg-45678abc
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}'
      InstanceType: !FindInMap [RegionAndEnvironmentToInstanceType, !Ref 'AWS::Region', !Ref EnvironmentType]
      SecurityGroupIds:
        - !FindInMap [RegionAndEnvironmentToSecurityGroup, !Ref 'AWS::Region', !Ref EnvironmentType]
```

Ressources connexes

Ces rubriques connexes peuvent vous être utiles pour développer des modèles utilisant la fonction `Fn::FindInMap`.

- [Fn::FindInMap](#)
- [Améliorations de Fn::FindInMap](#)
- [Fn::Sub](#)

CloudFormation Metadatasyntaxe du modèle

Metadata stocke des informations supplémentaires à l'aide d'objets JSON ou YAML. Les types de métadonnées au niveau du modèle que vous pouvez utiliser dans votre modèle sont les suivants :

Métadonnées personnalisées

Stocke des paires clé-valeur définies par l'utilisateur. Par exemple, vous pouvez fournir des informations supplémentaires qui n'ont pas d'incidence sur la création de ressources, mais qui offrent un contexte supplémentaire sur l'infrastructure, l'équipe ou les spécificités du déploiement.

`AWS::CloudFormation::Interface`

Définit le regroupement et l'ordre des paramètres d'entrée lorsqu'ils sont affichés dans la CloudFormation console. Par défaut, la CloudFormation console trie les paramètres par ordre alphabétique en fonction de leur identifiant logique.

`AWS::CloudFormation::Designer`

CloudFormation Designer (Designer) a atteint la fin de sa vie le 5 février 2025.

Important

Pendant une mise à jour de la pile, vous ne pouvez pas mettre à jour la section Metadata. Vous pouvez uniquement la mettre à jour lorsque vous intégrez des changements qui ajoutent, modifient ou suppriment des ressources.

CloudFormation ne transforme, ne modifie ni n'expurge aucune information que vous incluez dans Metadata cette section. Nous vous recommandons donc vivement de ne pas utiliser

cette section pour stocker des informations sensibles, telles que des mots de passe ou des secrets.

Syntaxe

Pour déclarer des métadonnées personnalisées dans votre CloudFormation modèle, utilisez la syntaxe suivante :

JSON

```
"Metadata" : {  
  "Instances" : {"Description" : "Information about the instances"},  
  "Databases" : {"Description" : "Information about the databases"}  
}
```

YAML

```
Metadata:  
  Instances:  
    Description: "Information about the instances"  
  Databases:  
    Description: "Information about the databases"
```

Pour la syntaxe du `AWS::CloudFormation::Interface`, voir [Organisation des CloudFormation paramètres à l'aide de AWS::CloudFormation::Interface métadonnées](#).

Organisation des CloudFormation paramètres à l'aide de **AWS::CloudFormation::Interface** métadonnées

`AWS::CloudFormation::Interface` est une clé de métadonnées qui définit la manière dont les paramètres sont regroupés et triés dans la CloudFormation console. Par défaut, la console répertorie les paramètres par ordre alphabétique de leurs identifiants logiques lorsque vous créez ou mettez à jour des piles. Cette clé vous permet de définir le regroupement et l'ordre de paramètres que vous souhaitez. De cette manière, les utilisateurs peuvent spécifier efficacement les valeurs des paramètres. Par exemple, vous pouvez regrouper tous les paramètres liés à EC2 dans un seul et même groupe et tous les paramètres liés au VPC dans un autre.

Dans la clé de métadonnées, vous pouvez spécifier les groupes à créer, les paramètres à inclure dans chaque groupe et l'ordre dans lequel la console affiche chaque paramètre au sein de son groupe.

Vous pouvez également définir des étiquettes pour les paramètres. Une étiquette est un nom descriptif ou une description que la console affiche au lieu de l'ID logique du paramètre. Les étiquettes sont utiles pour aider les utilisateurs à comprendre les valeurs à spécifier pour chaque paramètre. Par exemple, vous pouvez ajouter au paramètre `KeyPair` l'étiquette `Select an EC2 key pair`.

Tous les paramètres que vous référencez dans la clé de métadonnées doivent être déclarés dans la section `Parameters` du modèle.

Note

Seule la CloudFormation console utilise la clé de `AWS::CloudFormation::Interface` métadonnées. AWS CLI et les appels d'API n'utilisent pas cette clé.

Syntaxe

Pour déclarer cette entité dans votre CloudFormation modèle, utilisez la syntaxe suivante :

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups": [
      {
        "Label": {
          "default": "Group Label"
        },
        "Parameters": [
          "Parameter1",
          "Parameter2"
        ]
      }
    ],
    "ParameterLabels": {
      "Parameter1": {
        "default": "Friendly Name for Parameter1"
      }
    }
  }
}
```



```

    }
  }
}
}

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: Group Label
        Parameters:
          - Parameter1
          - Parameter2
    ParameterLabels:
      Parameter1:
        default: Friendly Name for Parameter1

```

Propriétés

ParameterGroups

Liste des types de groupes de paramètres, où vous spécifiez des noms de groupe, les paramètres de chaque groupe et l'ordre dans lequel les paramètres sont affichés.

Obligatoire : non

Label

Nom du groupe de paramètres.

Obligatoire : non

default

L'étiquette par défaut utilisée par la CloudFormation console pour nommer un groupe de paramètres.

Obligatoire : non

Type : chaîne

Parameters

Liste des paramètres logiques distinguant majuscules et minuscules IDs à inclure dans le groupe. Les paramètres doivent déjà être définis dans la section `Parameters` du modèle. Un paramètre ne peut être inclus que dans un seul groupe de paramètres.

La console répertorie les paramètres que vous n'associez pas à un groupe de paramètres dans le groupe `Other parameters`, dans l'ordre alphabétique.

Obligatoire : non

Type : liste de valeurs de chaîne

ParameterLabels

Un mappage des paramètres et de leurs noms conviviaux que la CloudFormation console affiche lors de la création ou de la mise à jour d'une pile.

Obligatoire : non

Étiquette de paramètre

Étiquette d'un paramètre. L'étiquette définit un nom ou une description convivial que la CloudFormation console affiche sur la page `Spécifier les paramètres` lorsqu'une pile est créée ou mise à jour. L'étiquette d'un paramètre doit correspondre à l'ID logique sensible à la casse d'un paramètre déclaré dans la section `Parameters` du modèle.

Obligatoire : non

`default`

L'étiquette par défaut utilisée par la CloudFormation console pour nommer un paramètre.

Obligatoire : non

Type : chaîne

Exemple

L'exemple suivant définit deux groupes de paramètres : `Network Configuration` et `Amazon EC2 Configuration`. Le groupe `Network Configuration` comprend les `VPCID`, `SubnetId` et

SecurityGroupID, qui sont définis dans la section Parameters du modèle (non affichée). L'ordre dans lequel la console affiche ces paramètres est défini par l'ordre dans lequel ils sont répertoriés, en commençant par le paramètre VPCID. Cet exemple regroupe et ordonne les paramètres Amazon EC2 Configuration d'une manière similaire.

Cet exemple définit également une étiquette pour le paramètre VPCID. La console affiche Which VPC should this be deployed to? (dans quel VPC ceci doit-il est déployé ?) au lieu de l'ID logique du paramètre (VPCID).

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups" : [
      {
        "Label" : { "default" : "Network Configuration" },
        "Parameters" : [ "VPCID", "SubnetId", "SecurityGroupID" ]
      },
      {
        "Label" : { "default":"Amazon EC2 Configuration" },
        "Parameters" : [ "InstanceType", "KeyName" ]
      }
    ],
    "ParameterLabels" : {
      "VPCID" : { "default" : "Which VPC should this be deployed to?" }
    }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Network Configuration"
        Parameters:
          - VPCID
          - SubnetId
          - SecurityGroupID
      - Label:
          default: "Amazon EC2 Configuration"
```

```

Parameters:
  - InstanceType
  - KeyName
ParameterLabels:
  VPCID:
    default: "Which VPC should this be deployed to?"

```

Groupes de paramètres dans la console

L'illustration suivante présente la façon dont la console affiche les groupes de paramètres lorsqu'une pile est créée ou mise à jour en utilisant la clé de métadonnées de cet exemple : Groupes de paramètres dans la console

Parameters

Network Configuration

Which VPC should this be deployed to?
VpcId of your existing Virtual Private Cloud (VPC)

SubnetId
The list of SubnetIds in your Virtual Private Cloud (VPC)

SecurityGroupId
VpcId of your existing Virtual Private Cloud (VPC)

Amazon EC2 Configuration

InstanceType Web

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instance

CloudFormation Rulesyntaxe du modèle

La `Rules` section est une partie facultative d'un CloudFormation modèle qui permet une logique de validation personnalisée. Lorsqu'elle est incluse, cette section contient des fonctions de règle qui valident les valeurs des paramètres avant que CloudFormation crée ou de mettre à jour des ressources.

Les règles sont utiles lorsque les contraintes standard des paramètres ne suffisent pas. Par exemple, lorsque SSL est activé, un certificat et un nom de domaine doivent être fournis. Une règle peut garantir que ces dépendances sont respectées.

Syntaxe

La section `Rules` utilise la syntaxe suivante :

JSON

La section `Rules` d'un modèle se compose du nom de clé `Rules`, suivi d'un seul signe deux points. Vous devez utiliser des accolades pour inclure toutes les déclarations de règle. Si vous déclarez plusieurs règles, elles sont délimitées par des virgules. Pour chaque règle, vous déclarez un nom logique entre guillemets, suivi d'un signe deux points et d'accolades qui entourent la condition de règle et les assertions.

```
{
  "Rules": {
    "LogicalRuleName1": {
      "RuleCondition": {
        "rule-specific intrinsic function": "Value"
      },
      "Assertions": [
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        },
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        }
      ]
    },
    "LogicalRuleName2": {
      "Assertions": [
        {
          "Assert": {
            "rule-specific intrinsic function": "Value"
          },
          "AssertDescription": "Information about this assert"
        }
      ]
    }
  }
}
```

```
}  
}
```

YAML

```
Rules:  
  LogicalRuleName1:  
    RuleCondition:  
      rule-specific intrinsic function: Value  
    Assertions:  
      - Assert:  
        rule-specific intrinsic function: Value  
        AssertDescription: Information about this assert  
      - Assert:  
        rule-specific intrinsic function: Value  
        AssertDescription: Information about this assert  
  LogicalRuleName2:  
    Assertions:  
      - Assert:  
        rule-specific intrinsic function: Value  
        AssertDescription: Information about this assert
```

Champs de règles

La section **Rules** peut inclure les champs suivants.

ID logique (également appelé nom logique)

Un identifiant unique pour chaque règle.

RuleCondition (facultatif)

Une propriété qui détermine quand une règle prend effet. Si vous ne définissez pas de condition de règle, les assertions de la règle prennent toujours effet. Pour chaque règle, vous ne pouvez définir qu'une seule condition de règle.

Assertions (obligatoire)

Une ou plusieurs instructions qui spécifient les valeurs acceptables pour un paramètre particulier.

Assert

Une condition qui doit être évaluée à `true`.

AssertDescription

Un message affiché lorsqu'une assertion échoue.

Fonctions intrinsèques spécifiques aux règles

Pour définir vos règles, vous devez utiliser des fonctions spécifiques aux règles, qui ne peuvent être utilisées que dans la section `Rules` d'un modèle. Bien que ces fonctions puissent être imbriquées, le résultat final d'une condition ou d'une assertion de règle doit être soit `true`, soit `false`.

Les fonctions de règle suivantes sont disponibles :

- [Fn::And](#)
- [Fn::Contains](#)
- [Fn::EachMemberEquals](#)
- [Fn::EachMemberIn](#)
- [Fn::Equals](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)
- [Fn::RefAll](#)
- [Fn::ValueOf](#)
- [Fn::ValueOfAll](#)

Ces fonctions sont utilisées dans la condition ou les assertions d'une règle. La propriété `condition` détermine si les CloudFormation assertions sont appliquées. Si la condition est évaluée à `true`, CloudFormation évalue les assertions pour vérifier si une valeur de paramètre est valide lors de la création ou de la mise à jour d'un produit approvisionné. Si une valeur de paramètre n'est pas valide, CloudFormation ne crée pas ou ne met pas à jour la pile. Si la condition est évaluée à `false`, CloudFormation elle ne vérifie pas la valeur du paramètre et poursuit l'opération de pile.

Exemples

Rubriques

- [Vérifier conditionnellement une valeur de paramètre](#)
- [Validation entre paramètres](#)

Vérifier conditionnellement une valeur de paramètre

Dans l'exemple suivant, les deux règles vérifient la valeur du paramètre `InstanceType`. En fonction de la valeur du paramètre d'environnement (`test` ou `prod`), l'utilisateur doit spécifier `t3.medium` ou `t3.large` pour le paramètre `InstanceType`. Les paramètres `InstanceType` et `Environment` doivent être déclarés dans la section `Parameters` du même modèle.

JSON

```
{
  "Rules": {
    "testInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "test"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.medium"],
              {"Ref": "InstanceType"}
            ]
          },
          "AssertDescription": "For a test environment, the instance type must be t3.medium"
        }
      ]
    },
    "prodInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "prod"
        ]
      },
      "Assertions": [
        {
          "Assert": {
            "Fn::Contains": [
              ["t3.large"],
```



```

        {"Ref": "InstanceType"}
      ]
    },
    "AssertDescription": "For a production environment, the instance type must be
t3.large"
  }
]
}
}
}
}

```

YAML

```

Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':
            - - t3.medium
              - !Ref InstanceType
          AssertDescription: 'For a test environment, the instance type must be
t3.medium'
      - prodInstanceType:
          RuleCondition: !Equals
            - !Ref Environment
            - prod
          Assertions:
            - Assert:
                'Fn::Contains':
                  - - t3.large
                    - !Ref InstanceType
                AssertDescription: 'For a production environment, the instance type must be
t3.large'

```

Validation entre paramètres

Les modèles d'exemple suivants illustrent l'utilisation des règles pour les validations interparamétriques. Ils créent un exemple de site web fonctionnant sur un groupe Auto Scaling derrière un équilibreur de charge. Le site web est accessible sur le port 80 ou 443, selon les

paramètres d'entrée. Les instances du groupe Auto Scaling peuvent être configurées pour écouter sur n'importe quel port (avec 8888 comme valeur par défaut).

Les règles de ce modèle valident les paramètres d'entrée avant la création de la pile. Elles vérifient que tous les sous-réseaux appartiennent au VPC spécifié et s'assurent que lorsque le paramètre UseSSL est défini sur Yes, un ARN de certificat SSL et un nom de zone hébergée sont fournis.

Note

Les AWS ressources utilisées vous seront facturées si vous créez une pile à partir de ce modèle.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "VpcId": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be the VPC Id of an existing Virtual Private Cloud."
    },
    "Subnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be a list of at least two existing subnets associated with at least two different availability zones."
    },
    "InstanceType": {
      "Description": "WebServer EC2 instance type",
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": ["t2.micro", "t3.micro"],
      "ConstraintDescription": "must be a valid EC2 instance type."
    },
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    }
  }
}
```

```

    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "UseSSL": {
      "AllowedValues": ["Yes", "No"],
      "Default": "No",
      "Description": "Select \"Yes\" to implement SSL, \"No\" to skip (default).",
      "Type": "String"
    },
    "ALBSSLCertificateARN": {
      "Default": "",
      "Description": "[Optional] The ARN of the SSL certificate to be used for the Application Load Balancer",
      "Type": "String"
    },
    "HostedZoneName": {
      "AllowedPattern": "^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-\\-]*[A-Za-z0-9])$",
      "Default": "",
      "Description": "[Optional] The domain name of a valid Hosted Zone on AWS.",
      "Type": "String"
    }
  },
  "Conditions": {
    "UseALBSSL": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]}
  },
  "Rules": {
    "SubnetsInVPC": {
      "Assertions": [
        {
          "Assert": {"Fn::EachMemberEquals": [{"Fn::ValueOf": ["Subnets", "VpcId"]}, {"Ref": "VpcId"}]},
          "AssertDescription": "All subnets must be in the VPC"
        }
      ]
    }
  }
}

```

```

    "ValidateHostedZone": {
      "RuleCondition": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]},
      "Assertions": [
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "ALBSSLCertificateARN"},
""]]}],
          "AssertDescription": "ACM Certificate value cannot be empty if SSL is
required"
        },
        {
          "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "HostedZoneName"}, ""]]}],
          "AssertDescription": "Route53 Hosted Zone Name is mandatory when SSL is
required"
        }
      ]
    }
  },
  "Resources": {
    "WebServerGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
        "VPCZoneIdentifier": {"Ref": "Subnets"},
        "LaunchTemplate": {
          "LaunchTemplateId": {"Ref": "LaunchTemplate"},
          "Version": {"Fn::GetAtt": ["LaunchTemplate", "LatestVersionNumber"]}
        },
        "MinSize": "2",
        "MaxSize": "2",
        "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
      },
      "CreationPolicy": {
        "ResourceSignal": {"Timeout": "PT15M"}
      },
      "UpdatePolicy": {
        "AutoScalingRollingUpdate": {
          "MinInstancesInService": "1",
          "MaxBatchSize": "1",
          "PauseTime": "PT15M",
          "WaitOnResourceSignals": true
        }
      }
    },
    "LaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",

```

```

"Metadata": {
  "Comment": "Install a simple application",
  "AWS::CloudFormation::Init": {
    "config": {
      "packages": {"yum": {"httpd": []}},
      "files": {
        "/var/www/html/index.html": {
          "content": {"Fn::Join": ["\n", ["<h1>Congratulations, you have
successfully launched the AWS CloudFormation sample.</h1>"]]},
          "mode": "000644",
          "owner": "root",
          "group": "root"
        },
        "/etc/cfn/cfn-hup.conf": {
          "content": {"Fn::Join": ["", [
            "[main]\n",
            "stack=", {"Ref": "AWS::StackId"}, "\n",
            "region=", {"Ref": "AWS::Region"}, "\n"
          ]]},
          "mode": "000400",
          "owner": "root",
          "group": "root"
        },
        "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
          "content": {"Fn::Join": ["", [
            "[cfn-auto-reloader-hook]\n",
            "triggers=post.update\n",
            "path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init\n",
            "action=/opt/aws/bin/cfn-init -v ",
            "          --stack ", {"Ref": "AWS::StackName"},
            "          --resource LaunchTemplate ",
            "          --region ", {"Ref": "AWS::Region"}, "\n",
            "runas=root\n"
          ]]},
          "mode": "000400",
          "owner": "root",
          "group": "root"
        }
      },
    },
    "services": {
      "sysvinit": {
        "httpd": {
          "enabled": "true",
          "ensureRunning": "true"
        }
      }
    }
  }
}

```

```

    },
    "cfn-hup": {
      "enabled": "true",
      "ensureRunning": "true",
      "files": [
        "/etc/cfn/cfn-hup.conf",
        "/etc/cfn/hooks.d/cfn-auto-reloader.conf"
      ]
    }
  }
},
"Properties": {
  "LaunchTemplateName": {"Fn::Sub": "${AWS::StackName}-launch-template"},
  "LaunchTemplateData": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "SecurityGroupIds": [{"Ref": "InstanceSecurityGroup"}],
    "InstanceType": {"Ref": "InstanceType"},
    "KeyName": {"Ref": "KeyName"},
    "UserData": {
      "Fn::Base64": {"Fn::Join": ["", [
        "#!/bin/bash\n",
        "yum install -y aws-cfn-bootstrap\n",
        "/opt/aws/bin/cfn-init -v ",
        "    --stack ", {"Ref": "AWS::StackName"},
        "    --resource LaunchTemplate ",
        "    --region ", {"Ref": "AWS::Region"}, "\n",
        "/opt/aws/bin/cfn-signal -e $? ",
        "    --stack ", {"Ref": "AWS::StackName"},
        "    --resource WebServerGroup ",
        "    --region ", {"Ref": "AWS::Region"}, "\n"
      ]]}
    }
  }
},
"ELBSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow access to the ELB",
    "VpcId": {"Ref": "VpcId"},

```

```

    "SecurityGroupIngress": [{
      "Fn::If": [
        "UseALBSSL",
        {
          "IpProtocol": "tcp",
          "FromPort": 443,
          "ToPort": 443,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        }
      ]
    }]
  },
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Subnets": {"Ref": "Subnets"},
      "SecurityGroups": [{"Ref": "ELBSecurityGroup"}]
    }
  },
  "ALBListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "DefaultActions": [{
        "Type": "forward",
        "TargetGroupArn": {"Ref": "ALBTargetGroup"}
      }],
      "LoadBalancerArn": {"Ref": "ApplicationLoadBalancer"},
      "Port": {"Fn::If": ["UseALBSSL", 443, 80]},
      "Protocol": {"Fn::If": ["UseALBSSL", "HTTPS", "HTTP"]},
      "Certificates": [{
        "Fn::If": [
          "UseALBSSL",
          {"CertificateArn": {"Ref": "ALBSSLCertificateARN"}},
          {"Ref": "AWS::NoValue"}
        ]
      }]
    }
  }
}

```

```

    },
    "ALBTargetGroup": {
      "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
      "Properties": {
        "HealthCheckIntervalSeconds": 30,
        "HealthCheckTimeoutSeconds": 5,
        "HealthyThresholdCount": 3,
        "Port": 80,
        "Protocol": "HTTP",
        "UnhealthyThresholdCount": 5,
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "SourceSecurityGroupId": {"Fn::Select": [0, {"Fn::GetAtt":
["ApplicationLoadBalancer", "SecurityGroups"]}]}
          },
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": {"Ref": "SSHLocation"}
          }
        ],
        "VpcId": {"Ref": "VpcId"}
      }
    },
    "RecordSet": {
      "Type": "AWS::Route53::RecordSetGroup",
      "Condition": "UseALBSSL",
      "Properties": {
        "HostedZoneName": {"Fn::Join": [ "", [{"Ref": "HostedZoneName"}, "." ] ] },
        "RecordSets": [ {
          "Name": {"Fn::Join": [ "", [
            {"Fn::Select": [ "0", {"Fn::Split": [ ".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]} ] ] ] ] ] },

```



```

        ". ",
        {"Ref": "HostedZoneName"},
        ". "
    ]],
    "Type": "A",
    "AliasTarget": {
        "DNSName": {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]},
        "EvaluateTargetHealth": true,
        "HostedZoneId": {"Fn::GetAtt": ["ApplicationLoadBalancer",
"CanonicalHostedZoneID"]}
    }
  ]
}
},
"Outputs": {
  "URL": {
    "Description": "URL of the website",
    "Value": {"Fn::Join": ["", [
      {"Fn::If": [
        "UseALBSSL",
        {"Fn::Join": ["", [
          "https://",
          {"Fn::Join": ["", [
            {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]},
            ". ",
            {"Ref": "HostedZoneName"},
            ". "
          ]}]
        ]}],
        {"Fn::Join": ["", [
          "http://",
          {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
        ]}]
      ]}
    ]}]
  }
}
}
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: must be the VPC Id of an existing Virtual Private Cloud.
  Subnets:
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
    ConstraintDescription: >-
      must be a list of at least two existing subnets associated with at least
      two different availability zones. They should be residing in the selected
      Virtual Private Cloud.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  UseSSL:
    AllowedValues:
      - 'Yes'
      - 'No'
    ConstraintDescription: Select Yes to create a HTTPS Listener
    Default: 'No'
    Description: 'Select "Yes" to implement SSL, "No" to skip (default).'
    Type: String
  ALBSSLCertificateARN:
```

```

    Default: ''
    Description: >-
        [Optional] The ARN of the SSL certificate to be used for the Application
        Load Balancer
    Type: String
    HostedZoneName:
        AllowedPattern: >-
            ^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9]|[A-Za-
z0-9][A-Za-z0-9\-\-]*[A-Za-z0-9])$
        Default: ''
        Description: '[Optional] The domain name of a valid Hosted Zone on AWS.'
        Type: String
    Conditions:
        UseALBSSL: !Equals
            - !Ref UseSSL
            - 'Yes'
    Rules:
        SubnetsInVPC:
            Assertions:
                - Assert:
                    'Fn::EachMemberEquals':
                        - 'Fn::ValueOf':
                            - Subnets
                            - VpcId
                        - Ref: VpcId
                    AssertDescription: All subnets must be in the VPC
        ValidateHostedZone:
            RuleCondition: !Equals
                - !Ref UseSSL
                - 'Yes'
            Assertions:
                - Assert: !Not
                    - !Equals
                    - !Ref ALBSSLCertificateARN
                    - ''
                    AssertDescription: ACM Certificate value cannot be empty if SSL is required
                - Assert: !Not
                    - !Equals
                    - !Ref HostedZoneName
                    - ''
                    AssertDescription: Route53 Hosted Zone Name is mandatory when SSL is required
    Resources:
        WebServerGroup:
            Type: AWS::AutoScaling::AutoScalingGroup

```

```
Properties:
  VPCZoneIdentifier: !Ref Subnets
  LaunchTemplate:
    LaunchTemplateId: !Ref LaunchTemplate
    Version: !GetAtt LaunchTemplate.LatestVersionNumber
  MinSize: '2'
  MaxSize: '2'
  TargetGroupARNs:
    - !Ref ALBTargetGroup
CreationPolicy:
  ResourceSignal:
    Timeout: PT15M
UpdatePolicy:
  AutoScalingRollingUpdate:
    MinInstancesInService: '1'
    MaxBatchSize: '1'
    PauseTime: PT15M
    WaitOnResourceSignals: 'true'
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
Metadata:
  Comment: Install a simple application
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
      files:
        /var/www/html/index.html:
          content: !Join
            - |+
            - - >-
              <h1>Congratulations, you have successfully launched the AWS
                CloudFormation sample.</h1>
          mode: '000644'
          owner: root
          group: root
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
          mode: '000400'
          owner: root
```

```

    group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
      content: !Sub |-
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
        runas=root
        mode: '000400'
        owner: root
        group: root
    services:
      sysvinit:
        httpd:
          enabled: 'true'
          ensureRunning: 'true'
        cfn-hup:
          enabled: 'true'
          ensureRunning: 'true'
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf

  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
          /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerGroup --region ${AWS::Region}
      ELBSecurityGroup:
        Type: AWS::EC2::SecurityGroup
        Properties:
          GroupDescription: Allow access to the ELB

```

```
VpcId: !Ref VpcId
SecurityGroupIngress:
  - !If
    - UseALBSSL
    - IpProtocol: tcp
      FromPort: 443
      ToPort: 443
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets: !Ref Subnets
    SecurityGroups:
      - !Ref ELBSecurityGroup
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: !If
      - UseALBSSL
      - 443
      - 80
    Protocol: !If
      - UseALBSSL
      - HTTPS
      - HTTP
    Certificates:
      - !If
        - UseALBSSL
        - CertificateArn: !Ref ALBSSLCertificateARN
        - !Ref 'AWS::NoValue'
ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 30
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 3
```

```

    Port: 80
    Protocol: HTTP
    UnhealthyThresholdCount: 5
    VpcId: !Ref VpcId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access and HTTP access on the inbound port
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId: !Select
          - 0
          - !GetAtt
            - ApplicationLoadBalancer
            - SecurityGroups
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
    VpcId: !Ref VpcId
RecordSet:
  Type: AWS::Route53::RecordSetGroup
  Condition: UseALBSSL
  Properties:
    HostedZoneName: !Join
      - ''
      - - !Ref HostedZoneName
      - .
  RecordSets:
    - Name: !Join
      - ''
      - - !Select
          - '0'
          - !Split
            - .
            - !GetAtt
              - ApplicationLoadBalancer
              - DNSName
        - .
        - !Ref HostedZoneName
        - .
    Type: A

```

```

AliasTarget:
  DNSName: !GetAtt
    - ApplicationLoadBalancer
    - DNSName
  EvaluateTargetHealth: true
  HostedZoneId: !GetAtt
    - ApplicationLoadBalancer
    - CanonicalHostedZoneID

Outputs:
  URL:
    Description: URL of the website
    Value: !Join
      - ''
      - - !If
          - UseALBSSL
          - !Join
            - ''
            - - 'https://'
              - !Join
                - ''
                - - !Select
                    - '0'
                    - !Split
                      - .
                      - !GetAtt
                        - ApplicationLoadBalancer
                        - DNSName
                  - .
                  - !Ref HostedZoneName
                - .
            - !Join
              - ''
              - - 'http://'
                - !GetAtt
                  - ApplicationLoadBalancer
                  - DNSName

```

CloudFormation Conditionsyntaxe du modèle

La section `Conditions` facultative contient des instructions qui définissent les circonstances dans lesquelles les entités sont créées et configurées. Par exemple, vous pouvez créer une condition et l'associer à une ressource ou à une sortie afin de CloudFormation créer la ressource ou la sortie uniquement si la condition est vraie. De même, vous pouvez associer une condition à une propriété

afin de CloudFormation définir une valeur spécifique à la propriété uniquement si la condition est vraie. Si la condition est fausse, CloudFormation définit la propriété sur une valeur alternative que vous spécifiez.

Vous pouvez utiliser des conditions lorsque vous souhaitez réutiliser un modèle pour créer des ressources dans différents contextes, comme les environnements de test et de production. Par exemple, dans votre modèle, vous pouvez ajouter un paramètre d'entrée `EnvironmentType` qui accepte soit `prod`, soit `test` comme valeurs d'entrée. Pour l'environnement `prod`, vous pourriez inclure des instances EC2 avec certaines capacités, tandis que pour l'environnement `test`, vous pourriez utiliser des capacités réduites afin d'économiser des coûts. Cette définition de condition vous permet de déterminer quelles ressources sont créées et comment elles sont configurées pour chaque type d'environnement.

Syntaxe

La section `Conditions` se compose du nom de clé `Conditions`. Chaque déclaration de condition comprend un ID logique et une ou plusieurs fonctions intrinsèques.

JSON

```
"Conditions": {  
  "LogicalConditionName1": {  
    "Intrinsic function": ...  
  },  
  
  "LogicalConditionName2": {  
    "Intrinsic function": ...  
  }  
}
```

YAML

```
Conditions:  
  LogicalConditionName1:  
    Intrinsic function:  
      ...  
  
  LogicalConditionName2:  
    Intrinsic function:  
      ...
```

Fonctionnement des conditions

Pour utiliser des conditions, procédez comme suit :

1. Ajouter une définition de paramètre : définissez les valeurs d'entrée que vos conditions devront évaluer dans la section `Parameters` du modèle. Les conditions s'évaluent comme « true » ou « false » en fonction de ces valeurs de paramètres d'entrée. Notez que les pseudo-paramètres sont automatiquement disponibles et ne nécessitent pas de définition explicite dans la section `Parameters`. Pour plus d'informations sur les pseudo-paramètres, consultez [Obtenir des AWS valeurs à l'aide de pseudo-paramètres](#).
2. Ajouter une définition de condition : définissez des conditions dans la section `Conditions` à l'aide de fonctions intrinsèques telles que `Fn::If` ou `Fn::Equals`. Ces conditions déterminent à quel moment CloudFormation les ressources associées sont créées. Les conditions peuvent se baser sur :
 - Valeurs de paramètres d'entrée ou de pseudo-paramètres
 - Autres conditions
 - Valeurs de mappage

Cependant, vous ne pouvez pas référencer la logique des ressources IDs ou leurs attributs dans certaines conditions.

3. Associer des conditions à des ressources ou à des sorties : référez les conditions dans les ressources ou les sorties à l'aide de la clé `Condition` et de l'ID logique d'une condition. Vous pouvez également utiliser `Fn::If` dans d'autres parties du modèle (comme des valeurs de propriétés) pour définir des valeurs dynamiques basées sur une condition. Pour de plus amples informations, veuillez consulter [Utilisation de la clé Condition](#).

CloudFormation évalue les conditions lors de la création ou de la mise à jour d'une pile.

CloudFormation crée des entités associées à une condition vraie et ignore les entités associées à une fausse condition. CloudFormation réévalue également ces conditions lors de chaque mise à jour de la pile avant de modifier les ressources. Les entités qui restent associées à une condition « true » sont mises à jour, tandis que celles associées à une condition « false » sont supprimées.

 Important

Pendant une mise à jour de la pile, vous ne pouvez pas mettre à jour les conditions. Vous pouvez uniquement les mettre à jour lorsque vous intégrez des changements qui ajoutent, modifient ou suppriment des ressources.

Fonctions intrinsèques de condition

Vous pouvez utiliser les fonctions intrinsèques suivantes pour définir des conditions :

- [Fn::And](#)
- [Fn::Equals](#)
- [Fn::ForEach](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)

 Note

Fn::If n'est pris en charge que dans l'attribut de métadonnées, l'attribut de politique de mise à jour et les valeurs de propriété dans les sections Ressources et Outputs d'un modèle.

Utilisation de la clé **Condition**

Une fois une condition définie, vous pouvez l'appliquer à plusieurs endroits du modèle, comme dans Ressources et Outputs, à l'aide de la clé Condition. La clé Condition référence le nom logique d'une condition et renvoie le résultat évalué de la condition spécifiée.

Rubriques

- [Association des conditions aux ressources](#)
- [Association des conditions aux sorties](#)
- [Référencement des conditions dans d'autres conditions](#)

- [Renvoi conditionnel de valeurs de propriétés à l'aide de Fn::If](#)

Association des conditions aux ressources

Pour créer des ressources de manière conditionnelle, ajoutez la `Condition` clé et l'ID logique de la condition en tant qu'attribut à la ressource. CloudFormation crée la ressource uniquement lorsque la condition est considérée comme vraie.

JSON

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Condition" : "IsProduction",
  "Properties" : {
    "Size" : "100",
    "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ] }
  }
}
```

YAML

```
NewVolume:
  Type: AWS::EC2::Volume
  Condition: IsProduction
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone
```

Association des conditions aux sorties

Vous pouvez également associer des conditions aux sorties. CloudFormation crée la sortie uniquement lorsque la condition associée est considérée comme vraie.

JSON

```
"Outputs" : {
  "VolumeId" : {
    "Condition" : "IsProduction",
    "Value" : { "Ref" : "NewVolume" }
  }
}
```

```
}
```

YAML

```
Outputs:
  VolumeId:
    Condition: IsProduction
    Value: !Ref NewVolume
```

Référencement des conditions dans d'autres conditions

Lorsque vous définissez des conditions dans la section `Conditions`, vous pouvez référencer d'autres conditions à l'aide de la clé `Condition`. Cela vous permet de créer une logique conditionnelle plus complexe en combinant plusieurs conditions.

Dans l'exemple suivant, la condition `IsProdAndFeatureEnabled` s'évalue à « true » uniquement si les conditions `IsProduction` et `IsFeatureEnabled` s'évaluent toutes deux à « true ».

JSON

```
"Conditions": {
  "IsProduction" : {"Fn::Equals" : [{"Ref" : "Environment"}, "prod"]},
  "IsFeatureEnabled" : { "Fn::Equals" : [{"Ref" : "FeatureFlag"}, "enabled"]},
  "IsProdAndFeatureEnabled" : {
    "Fn::And" : [
      {"Condition" : "IsProduction"},
      {"Condition" : "IsFeatureEnabled"}
    ]
  }
}
```

YAML

```
Conditions:
  IsProduction:
    !Equals [!Ref Environment, "prod"]
  IsFeatureEnabled:
    !Equals [!Ref FeatureFlag, "enabled"]
  IsProdAndFeatureEnabled: !And
    - !Condition IsProduction
    - !Condition IsFeatureEnabled
```

Renvoi conditionnel de valeurs de propriétés à l'aide de **Fn::If**

Pour un contrôle plus précis, vous pouvez utiliser la fonction intrinsèque `Fn::If` afin de renvoyer conditionnellement l'une de deux valeurs dans des propriétés de ressources ou de sorties. Cette fonction évalue une condition et renvoie une valeur si la condition est « true », et une autre valeur si la condition est « false ».

Valeurs conditionnelles de propriétés

L'exemple suivant montre comment définir le type d'instance EC2 en fonction d'une condition liée à l'environnement. Si la condition `IsProduction` s'évalue à « true », le type d'instance est défini sur `c5.xlarge`. Sinon, elle est définie sur `t3.small`.

JSON

```
"Properties" : {
  "InstanceType" : {
    "Fn::If" : [
      "IsProduction",
      "c5.xlarge",
      "t3.small"
    ]
  }
}
```

YAML

```
Properties:
  InstanceType: !If
    - IsProduction
    - c5.xlarge
    - t3.small
```

Suppression conditionnelle de propriétés

Vous pouvez également utiliser le pseudo-paramètre `AWS::NoValue` comme valeur de renvoi pour supprimer la propriété correspondante lorsqu'une condition s'évalue à « false ».

JSON

```
"DBSnapshotIdentifier" : {
  "Fn::If" : [
```

```
    "UseDBSnapshot",
    {"Ref" : "DBSnapshotName"},
    {"Ref" : "AWS::NoValue"}
  ]
}
```

YAML

```
DBSnapshotIdentifier: !If
- UseDBSnapshot
- !Ref DBSnapshotName
- !Ref "AWS::NoValue"
```

Exemples

Rubriques

- [Création de ressources en fonction de l'environnement](#)
- [Provisionnement de ressources basé sur plusieurs conditions](#)

Création de ressources en fonction de l'environnement

Les exemples suivants provisionnent une instance EC2 et créent puis attachent un volume EBS supplémentaire uniquement si le type d'environnement est prod. Si l'environnement est test, seule l'instance EC2 est créée, sans volume supplémentaire.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Description": "Environment type",
      "Default": "test",
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ],
      "ConstraintDescription": "must specify prod or test"
    }
  },
}
```

```
"Conditions": {
  "IsProduction": {
    "Fn::Equals": [
      {
        "Ref": "EnvType"
      },
      "prod"
    ]
  }
},
"Resources": {
  "EC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "ami-1234567890abcdef0",
      "InstanceType": "c5.xlarge"
    }
  },
  "MountPoint": {
    "Type": "AWS::EC2::VolumeAttachment",
    "Condition": "IsProduction",
    "Properties": {
      "InstanceId": {
        "Ref": "EC2Instance"
      },
      "VolumeId": {
        "Ref": "NewVolume"
      },
      "Device": "/dev/sdh"
    }
  },
  "NewVolume": {
    "Type": "AWS::EC2::Volume",
    "Condition": "IsProduction",
    "Properties": {
      "Size": 100,
      "AvailabilityZone": {
        "Fn::GetAtt": [
          "EC2Instance",
          "AvailabilityZone"
        ]
      }
    }
  }
}
```



```
}  
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09  
Parameters:  
  EnvType:  
    Description: Environment type  
    Default: test  
    Type: String  
    AllowedValues:  
      - prod  
      - test  
    ConstraintDescription: must specify prod or test  
Conditions:  
  IsProduction: !Equals  
    - !Ref EnvType  
    - prod  
Resources:  
  EC2Instance:  
    Type: AWS::EC2::Instance  
    Properties:  
      ImageId: ami-1234567890abcdef0  
      InstanceType: c5.xlarge  
  MountPoint:  
    Type: AWS::EC2::VolumeAttachment  
    Condition: IsProduction  
    Properties:  
      InstanceId: !Ref EC2Instance  
      VolumeId: !Ref NewVolume  
      Device: /dev/sdh  
  NewVolume:  
    Type: AWS::EC2::Volume  
    Condition: IsProduction  
    Properties:  
      Size: 100  
      AvailabilityZone: !GetAtt  
        - EC2Instance  
        - AvailabilityZone
```

Provisionnement de ressources basé sur plusieurs conditions

Les exemples suivants créent conditionnellement un compartiment S3 lorsqu'un nom de compartiment est fourni, puis attachent une stratégie de compartiment uniquement lorsque l'environnement est défini sur prod. Si aucun nom de compartiment n'est fourni ou si l'environnement est test, aucune ressource n'est créée.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ]
    },
    "BucketName": {
      "Default": "",
      "Type": "String"
    }
  },
  "Conditions": {
    "IsProduction": {
      "Fn::Equals": [
        {
          "Ref": "EnvType"
        },
        "prod"
      ]
    },
    "CreateBucket": {
      "Fn::Not": [
        {
          "Fn::Equals": [
            {
              "Ref": "BucketName"
            },
            ""
          ]
        }
      ]
    }
  }
}
```

```

    ]
  },
  "CreateBucketPolicy": {
    "Fn::And": [
      {
        "Condition": "IsProduction"
      },
      {
        "Condition": "CreateBucket"
      }
    ]
  }
},
"Resources": {
  "Bucket": {
    "Type": "AWS::S3::Bucket",
    "Condition": "CreateBucket",
    "Properties": {
      "BucketName": {
        "Ref": "BucketName"
      }
    }
  },
  "Policy": {
    "Type": "AWS::S3::BucketPolicy",
    "Condition": "CreateBucketPolicy",
    "Properties": {
      "Bucket": {
        "Ref": "Bucket"
      },
      "PolicyDocument": { ... }
    }
  }
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Type: String
    AllowedValues:

```

```
- prod
- test
BucketName:
  Default: ''
  Type: String
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
  CreateBucket: !Not
    - !Equals
      - !Ref BucketName
      - ''
  CreateBucketPolicy: !And
    - !Condition IsProduction
    - !Condition CreateBucket
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Condition: CreateBucket
    Properties:
      BucketName: !Ref BucketName
  Policy:
    Type: AWS::S3::BucketPolicy
    Condition: CreateBucketPolicy
    Properties:
      Bucket: !Ref Bucket
      PolicyDocument: ...
```

Dans cet exemple, la condition `CreateBucketPolicy` montre comment référencer d'autres conditions à l'aide de la clé `Condition`. La stratégie est créée uniquement lorsque les conditions `IsProduction` et `CreateBucket` s'évaluent toutes deux à « true ».

Note

Pour des exemples plus complexes d'utilisation des conditions, consultez la rubrique [Attribut Condition](#) dans le Guide de référence des modèles CloudFormation .

CloudFormation Transform section des modèles

La Transform section facultative indique une ou plusieurs macros CloudFormation utilisées pour traiter votre modèle d'une manière ou d'une autre.

Les macros peuvent effectuer des tâches simples telles que rechercher et remplacer du texte, ou elles peuvent apporter des transformations plus importantes à l'ensemble du modèle.

CloudFormation exécute les macros dans l'ordre dans lequel elles sont spécifiées. Lorsque vous créez un ensemble de modifications, il en CloudFormation génère un qui inclut le contenu du modèle traité. Vous pouvez alors passer en revue les modifications et exécuter le jeu de modifications. Pour plus d'informations sur le fonctionnement des macros, consultez [Effectuez un traitement personnalisé sur les CloudFormation modèles à l'aide de macros de modèles](#).

CloudFormation prend également en charge les transformations, qui sont des macros hébergées par CloudFormation. CloudFormation traite ces transformations de la même manière que toutes les macros que vous créez en termes d'ordre d'exécution et de portée. Pour plus d'informations, consultez [Référence des transformations](#).

Pour déclarer plusieurs macros, utilisez un format de liste et spécifiez une ou plusieurs macros.

Par exemple, dans l'exemple de modèle ci-dessous, CloudFormation évalue MyMacro puis AWS::Serverless traite le contenu de l'ensemble du modèle en raison de son inclusion dans la Transform section.

```
# Start of processable content for MyMacro and AWS::Serverless
Transform:
  - MyMacro
  - 'AWS::Serverless'
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key": "value"}]
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-1234567890abcdef0
```

```
# End of processable content for MyMacro and AWS::Serverless
```

CloudFormation syntaxe de version du format de modèle

La section `AWSTemplateFormatVersion` (facultative) identifie la version du format de modèle auquel le modèle est conforme. La dernière version de format du modèle est `2010-09-09`. Il s'agit actuellement de la seule valeur valide.

La version du format du modèle n'est pas la même que la version de l'API. La version du format du modèle peut changer indépendamment des versions de l'API.

La valeur de la déclaration de la version de format du modèle doit être une chaîne littérale. Vous ne pouvez pas utiliser une fonction ou un paramètre pour spécifier la version de format du modèle. Si vous ne spécifiez aucune valeur, il s'agit CloudFormation de la dernière version du format du modèle. L'extrait suivant présente un exemple de déclaration valide d'une version de format d'un modèle :

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
```

CloudFormation Description syntaxe du modèle

La section `Description` (facultative) vous permet d'inclure une chaîne de texte qui décrit le modèle. Cette section doit toujours suivre la section de version du format du modèle.

La valeur de la déclaration de la description doit être une chaîne littérale dont la taille doit être comprise entre 0 et 1 024 octets. Vous ne pouvez pas utiliser une fonction ou un paramètre pour spécifier la description. L'extrait suivant est un exemple de déclaration de description :

Important

Pendant une mise à jour de la pile, vous ne pouvez pas mettre à jour la section `Description`. Vous pouvez uniquement la mettre à jour lorsque vous intégrez des changements qui ajoutent, modifient ou suppriment des ressources.

JSON

```
"Description" : "Here are some details about the template."
```

YAML

```
Description: > Here are some details about the template.
```

Créez des modèles visuellement avec Infrastructure Composer

AWS Infrastructure Composer (anciennement connu sous le nom d'Application Composer) vous permet de composer et de configurer visuellement des applications modernes sur AWS. Au lieu d'écrire du code, vous pouvez glisser-déposer différentes ressources pour créer visuellement votre application.

Infrastructure Composer en mode CloudFormation console est l'outil recommandé pour travailler visuellement avec CloudFormation des modèles. Cette version d'Infrastructure Composer à laquelle vous pouvez accéder depuis la CloudFormation console est une amélioration par rapport à un ancien outil appelé CloudFormation Designer.

Avec Infrastructure Composer en mode CloudFormation console, vous pouvez glisser-déposer, configurer et connecter diverses ressources, appelées cartes, sur un canevas visuel. Cette approche visuelle facilite la conception et la modification de l'architecture de votre application sans avoir à travailler directement avec des modèles. Pour accéder à ce mode depuis la [CloudFormation console](#), sélectionnez Infrastructure Composer dans le menu de navigation de gauche.

Pour plus d'informations, consultez la section [Comment composer dans AWS Infrastructure Composer](#) dans le Guide du développeur AWS Infrastructure Composer .

Pourquoi utiliser Infrastructure Composer en mode CloudFormation console ?

La visualisation de vos modèles dans Infrastructure Composer vous permet d'identifier les lacunes et les domaines à améliorer dans vos CloudFormation modèles et dans l'architecture de vos applications. Infrastructure Composer améliore votre expérience de développement grâce à la facilité et à l'efficacité de la création et de la modification visuelles de CloudFormation piles. Vous

pouvez commencer par une ébauche initiale, créer un code déployable et intégrer vos workflows de développement à l'aide du concepteur visuel dans Infrastructure Composer.

En quoi ce mode est-il différent de celui de la console Infrastructure Composer ?

Bien que la version CloudFormation console d'Infrastructure Composer présente des fonctionnalités similaires à celles de la console Infrastructure Composer standard, il existe quelques différences. Les cartes associées à Lambda (fonction Lambda et couche Lambda) nécessitent des compilations de code et des solutions de packaging qui ne sont pas disponibles dans Infrastructure Composer en mode console. CloudFormation La synchronisation locale n'est pas non plus disponible dans ce mode.

Cependant, vous pouvez utiliser ces cartes liées à Lambda et la fonctionnalité de synchronisation locale dans la [console Infrastructure Composer](#) ou le AWS Toolkit for Visual Studio Code.

Pour plus d'informations, consultez le [Guide du développeur AWS Infrastructure Composer](#) et [Infrastructure Composer](#) dans le Guide de l'utilisateur AWS Toolkit for Visual Studio Code .

AWS CloudFormation Serveur de langues

Le serveur de AWS CloudFormation langue fournit des fonctionnalités permettant d'accélérer la création infrastructure-as-code (iAc) et le déploiement AWS des ressources en toute sécurité et en toute confiance. Il suit le [protocole du serveur de langage](#) (LSP) pour fournir de la documentation sur le survol, l'auto-complétion, les diagnostics via la validation statique, le passage à la définition et les actions de code. Outre ces fonctionnalités de serveur linguistique traditionnel, le serveur ajoute des fonctionnalités en ligne permettant d'explorer et de déployer AWS des ressources via CloudFormation. Cela inclut la possibilité de valider et de déployer des modèles à l'aide d'ensembles de modifications, d'afficher les différences de pile, les événements, les ressources et les résultats, de répertorier les piles et de parcourir les ressources par type, et d'insérer l'état des ressources en direct directement dans CloudFormation les modèles.

Rubriques

- [IDEs intégration avec le serveur de AWS CloudFormation langue](#)
- [Prise en main](#)
- [Initialisation d'un CloudFormation projet dans l'IDE \(VS Code uniquement\)](#)
- [Open source](#)
- [Vous avez besoin d'aide ?](#)

IDEs intégration avec le serveur de AWS CloudFormation langue

AWS fournit une off-the-shelf intégration avec le serveur de CloudFormation langue via le AWS kit d'outils pour les éléments suivants IDEs :

- [Code Visual Studio](#)
- [JetBrains IDEs](#)(version 2025.3 ou ultérieure), y compris IntelliJ IDEA, et WebStorm PyCharm

Les versions suivantes prennent IDEs également en charge le serveur de CloudFormation langue :

- [Kiro](#)
- [Curseur](#)
- La plupart des forks et distributions de VS Code

Le serveur de CloudFormation langue adhère au protocole LSP ([Language Server Protocol](#)) et les autres intégrations sont donc configurables. Pour obtenir des instructions sur l'intégration du serveur de langue à d'autres éditeurs, consultez le [guide d'installation](#).

Prise en main

Rubriques

- [Conditions préalables](#)
- [Étape 1 : Installation ou mise à niveau du AWS kit d'outils](#)
- [Étape 2 : Accès CloudFormation dans la AWS boîte à outils](#)
- [Étape 3 : Validez, testez et affinez votre modèle](#)
- [Étape 4 : Naviguez dans le modèle](#)
- [Étape 5 : Valider et déployer](#)

Conditions préalables

Avant de commencer, assurez-vous que :

- Vous utilisez un IDE compatible sur un système d'exploitation compatible (macOS, Windows, ou Linux).

- Vous avez installé ou effectué une mise à niveau vers la dernière version du AWS Toolkit pour votre IDE.

Certaines fonctionnalités du serveur de AWS CloudFormation langue nécessitent des informations d'identification actives Compte AWS et configurées. Vous devez être connecté à votre compte Compte AWS via le AWS Toolkit à l'aide d'informations d'identification valides.

Étape 1 : Installation ou mise à niveau du AWS kit d'outils

Installez ou mettez à jour la dernière version du AWS Toolkit à partir du gestionnaire d'extensions ou de plugins de votre IDE, puis redémarrez votre IDE.

Après l'installation, le AWS Toolkit active automatiquement le support CloudFormation IDE. Lorsque vous installez ou mettez à niveau le AWS kit d'outils avec le serveur de AWS CloudFormation langue pour la première fois, vous êtes invité AWS à autoriser la collecte de données d'utilisation anonymes. Ces données permettent AWS d'améliorer le serveur de CloudFormation langue et d'améliorer l'expérience de création. Aucune information sensible n'est collectée et AWS n'enregistre ni ne stocke le contenu des modèles, les configurations des ressources ou les données clients identifiables. Vous pouvez modifier vos préférences de télémétrie à tout moment depuis les paramètres de l'IDE. Redémarrez l'IDE pour que les modifications prennent effet. Les données d'utilisation collectées se concentrent uniquement sur les interactions entre les fonctionnalités et les indicateurs de performance. Ces informations permettent AWS d'identifier et de hiérarchiser les améliorations telles qu'une validation plus rapide, une saisie semi-automatique améliorée et un meilleur diagnostic des erreurs.

Étape 2 : Accès CloudFormation dans la AWS boîte à outils

Après avoir installé le AWS Toolkit, ouvrez le CloudFormation panneau dans votre IDE. Dans VS Code, ouvrez le panneau AWS Toolkit depuis la barre d'activité et choisissez CLOUDFORMATION. Dans JetBrains IDEs, ouvrez la fenêtre de l'outil AWS Toolkit depuis la barre latérale et sélectionnez l'CloudFormationonglet.

Le CloudFormation panneau contient les sections suivantes :

- Région : affiche le courant Région AWS. Dans VS Code, vous pouvez le modifier en sélectionnant le nom de la région ou en utilisant la commande AWS CloudFormation: Sélectionner une région dans la palette de commandes. Dans JetBrains IDEs, la région est configurée via les paramètres de connexion du AWS Toolkit.

- **Piles** : affiche une liste paginée des CloudFormation piles de votre compte. Développez une pile pour afficher son nœud Change Sets, qui répertorie les ensembles de modifications associés à cette pile. Utilisez l'action Afficher les détails de la pile pour ouvrir la vue détaillée de la pile, qui affiche la vue d'ensemble de la pile, les événements, les sorties et les ressources.
- **Ressources** : une fois que vous avez ajouté un type de ressource, le panneau affiche les AWS ressources de ce type dans votre compte. Vous pouvez les afficher, les actualiser, les copier ou les importer dans votre modèle.

Dans JetBrains IDEs, la barre d'outils située au-dessus de l'arborescence permet d'accéder rapidement aux actions courantes, notamment valider et déployer, réexécuter la validation et le déploiement, ajouter un type de ressource et actualiser. Les actions sont également disponibles via les menus contextuels du bouton droit sur les nœuds de l'arborescence.

Étape 3 : Validez, testez et affinez votre modèle

Lorsque vous rédigez votre CloudFormation modèle, l'IDE fournit une assistance intelligente à la création pour vous aider à créer plus rapidement une infrastructure précise et conforme. Le serveur de CloudFormation langue s'exécute en arrière-plan et fournit les fonctionnalités de création suivantes :

- **Complétion du code** : suggère des types de ressources, des paramètres et des propriétés en fonction de CloudFormation schémas.
- **Ajouter AWS des ressources existantes** : vous permet d'importer des ressources existantes depuis votre Compte AWS modèle. L'IDE utilise le [API de commande du Cloud AWS \(CCAPI\)](#) pour récupérer la configuration en direct et les propriétés de la ressource, afin de vous aider à cloner ou à réutiliser l'infrastructure existante dans votre modèle.
- **Extraire vers le paramètre** : lorsque votre curseur se trouve sur une valeur littérale dans un modèle (par exemple, une chaîne de caractères `2.micro`), l'IDE propose une action de refactorisation pour extraire la valeur dans la `Parameters` section et remplacer le littéral par un `!Ref` pour le nouveau paramètre. Si la même valeur littérale apparaît à plusieurs endroits, vous pouvez choisir d'extraire toutes les occurrences en une seule fois.

Pour ajouter des ressources à votre modèle

- **Ajouter un type de ressource** : dans le CloudFormation panneau AWS Boîte à outils, sous Ressources, ajoutez un type de ressource à parcourir. Dans VS Code, cliquez sur l'icône Ajouter + ou utilisez la commande AWS CloudFormation: Ajouter des types de ressources dans la palette

de commandes. Dans JetBrains, cliquez sur le bouton Ajouter un type de ressource dans la barre d'outils ou cliquez avec le bouton droit sur le nœud Ressources.

- Rechercher un type de ressource : dans la boîte de dialogue de recherche, tapez le type de AWS ressource que vous souhaitez ajouter. Exemple :
 - `AWS::S3::Bucket`
 - `AWS::Lambda::Function`
- Parcourir les ressources : dans la section Ressources, une liste paginée des AWS ressources détectées dans votre compte s'affiche. Si vous disposez de nombreuses ressources, seule la première page est affichée. Utilisez les commandes de navigation pour parcourir les pages supplémentaires et consulter toutes les ressources disponibles.
- Choisissez la ressource que vous souhaitez inclure dans votre modèle.
- Vous pouvez insérer une ressource dans votre modèle de deux manières, en fonction de votre objectif :
 - Cloner une ressource existante : créez une nouvelle ressource dans votre modèle en utilisant la configuration en direct et les propriétés d'une AWS ressource existante.
 - Importer une ressource existante : insérez la ressource réelle dans votre pile en l'ajoutant à votre modèle en utilisant son état actif.

Astuces

- Vous pouvez actualiser la section Ressources à tout moment pour consulter la dernière liste des ressources disponibles dans votre compte ou dans votre région.
- Si vous importez des ressources, n'ajoutez pas de ressource qui appartient déjà à une CloudFormation pile existante dans le même compte.
- Pour vérifier si une ressource est déjà gérée par CloudFormation, utilisez l'action d'information située à côté de la ressource. Dans VS Code, cliquez sur l'icône . Dans JetBrains IDEs, cliquez avec le bouton droit sur la ressource et choisissez Get Stack Management Info.

Ajouter des ressources connexes

Dans VS Code, vous pouvez ajouter des ressources associées à la ressource sélectionnée à l'aide de la commande AWS CloudFormation: Ajouter des ressources associées par type. Une fois que vous avez sélectionné un type de ressource parmi ceux déjà définis dans votre modèle, l'IDE affiche une liste de ressources généralement associées ou dépendantes de ce type. Par exemple, si vous sélectionnez un `AWS::EC2::Instance`, l'IDE peut suggérer d'ajouter des ressources connexes

telles que `AWS::EC2::SecurityGroup` ou `AWS::EC2::Subnet`. Cette fonctionnalité vous permet de créer rapidement des composants d'infrastructure connectés sans rechercher manuellement des types de ressources compatibles. Cette fonctionnalité n'est actuellement pas prise en charge dans JetBrains IDEs.

Validation statique

Le serveur de CloudFormation langue fournit une validation statique intégrée alimentée par [AWS CloudFormation Linter \(cfn-lint\)](#) et [AWS CloudFormation Guard](#). Ces validations s'exécutent en arrière-plan lorsque vous créez des modèles, ce qui vous aide à identifier les erreurs de syntaxe, les lacunes de conformité et les problèmes liés aux meilleures pratiques avant le déploiement.

Vue d'ensemble de la validation statique

Vous verrez deux types de validations statiques en temps réel dans l'IDE :

- CloudFormation Linter (`cfn-lint`) : valide votre modèle par rapport aux spécifications CloudFormation des ressources et aux règles du schéma.
- Guard (`cfn-guard`) : valide votre modèle par rapport aux règles de conformité et aux packs de politiques organisationnelles.

CloudFormation Linter (cfn-lint)

Le CloudFormation Linter est intégré à l'IDE pour vérifier automatiquement la syntaxe et la structure de votre modèle au fur et à mesure que vous tapez.

- Validation du schéma : détecte les erreurs de syntaxe et de schéma pour garantir que vos modèles sont conformes au schéma CloudFormation des ressources.
- Surlignage des erreurs : affiche des marqueurs intégrés sous les problèmes, représentant les bloqueurs de déploiement ou les avertissements.
- Aide au survol : lorsque vous survolez une erreur, l'IDE affiche le message de diagnostic associé à ce problème. Si une solution rapide est disponible, elle est également proposée.

Intégration de Guard

Guard valide vos modèles par rapport à des ensembles de règles qui définissent les politiques de conformité et de sécurité. L'IDE exécute les validations Guard en temps réel via le serveur de CloudFormation langue, ce qui vous permet de recevoir des commentaires immédiats lorsque vous créez des modèles.

- Packs de règles par défaut : l'IDE inclut un ensemble préenregistré de règles Guard axées sur les meilleures pratiques fondamentales en matière de sécurité des ressources et d'hygiène de configuration. Pour en savoir plus, consultez [le registre des règles de garde](#).
- Ajout de packs de règles : pour ajouter ou modifier des ensembles de règles, ouvrez les paramètres de votre IDE et accédez à la section de configuration de Guard pour sélectionner ou télécharger des packs de règles Guard supplémentaires.

Conseils : Comprendre les indicateurs diagnostiques

- Indicateurs bleus : conseils sur les meilleures pratiques ou recommandations d'optimisation.
- Indicateurs jaunes : avertissements concernant des problèmes non bloquants (par exemple, des balises ou des paramètres manquants).
- Indicateurs rouges : bloqueurs de déploiement tels que des noms de propriété non valides, des champs obligatoires manquants ou des incohérences de schéma.

Étape 4 : Naviguez dans le modèle

L'IDE fournit une vue hiérarchique structurée de votre CloudFormation modèle, organisée en sections telles que `Parameters`, `Resources`, et `OutputsMappings`, indiquant chaque type de ressource et chaque identifiant logique. Cela permet de localiser et de naviguer rapidement vers des ressources ou des paramètres spécifiques dans de grands modèles. Dans VS Code, le panneau Outline de la barre latérale de l'explorateur affiche cette structure. Dans JetBrains IDEs, ouvrez la fenêtre de l'outil Structure pour afficher la structure du modèle du fichier actuellement ouvert.

Vous pouvez utiliser Go to Definition pour des fonctions intrinsèques telles que `GetAtt` et `Ref`, ce qui vous permet d'accéder directement à la ressource ou au paramètre référencé dans votre modèle. Cela vous permet de suivre les dépendances, de comprendre les relations entre les ressources et d'apporter des modifications plus efficacement.

Étape 5 : Valider et déployer

Lorsque vous êtes prêt à déployer votre CloudFormation modèle, utilisez la fonctionnalité de validation et de déploiement pour créer un ensemble de modifications. L'IDE valide votre modèle et, si aucune erreur de blocage n'est détectée, il crée un ensemble de modifications adapté à la [dérive](#). L'IDE affiche ensuite une vue diff afin que vous puissiez passer en revue toutes les modifications proposées avant d'exécuter l'ensemble de modifications.

Dans VS Code, ouvrez la palette de commandes et exécutez AWS CloudFormation: Valider et déployer. La palette de commandes vous guide étape par étape dans la sélection d'un modèle, d'un nom de pile, de paramètres, de fonctionnalités et d'autres options de déploiement. Dans JetBrains IDEs, utilisez le bouton Valider et déployer de la barre d'outils, cliquez avec le bouton droit sur un fichier modèle dans l'éditeur ou cliquez avec le bouton droit sur une pile dans l'arborescence. JetBrains présente une boîte de dialogue d'assistant dans laquelle vous configurez toutes les options de déploiement, y compris la sélection du modèle, le nom de la pile, les paramètres, les fonctionnalités, les balises et les options avancées.

Comment fonctionne la validation

L'IDE effectue automatiquement un [contrôle de validation avant le déploiement](#) et valide votre modèle par rapport aux causes de défaillance courantes, notamment :

- Syntaxe de propriété non valide ou incompatibilité du schéma : ces problèmes sont généralement détectés `cfn-lint` lors de la création, mais si vous procédez au déploiement sans les résoudre, la validation au moment CloudFormation du déploiement fait apparaître les mêmes erreurs avant la création ou la mise à jour de la pile.
- Le nom de la ressource est en conflit avec les ressources existantes de votre compte.
- Contraintes spécifiques au service, telles que les conflits de noms de compartiment S3 ou l'absence de chiffrement.

Si la validation détecte des erreurs, l'IDE met en évidence les problèmes directement dans votre modèle et répertorie les erreurs dans le panneau de diagnostic. Chaque problème inclut la propriété ou la ressource spécifique à l'origine de l'échec, ainsi qu'une solution suggérée. S'il n'y a aucune erreur de blocage, vous pouvez passer à la phase de déploiement.

Si des avertissements sont détectés (problèmes non bloquants), une boîte de dialogue apparaît vous permettant de poursuivre le déploiement ou d'annuler et d'apporter des corrections.

L'IDE ouvre un [ensemble de modifications sensible à la dérive](#) qui affiche les différences entre votre modèle actuel et la configuration de la pile déployée. Cela vous permet de revoir, de confirmer ou d'annuler la modification définie avant son exécution. L'annulation du déploiement entraîne la suppression de l'ensemble de modifications.

Les ensembles de modifications sensibles à la dérive améliorent le processus de CloudFormation déploiement en vous permettant de gérer la dérive des piles en toute sécurité. La dérive de pile se produit lorsque l'état réel de vos ressources diffère de ce qui est défini dans votre CloudFormation

modèle, souvent en raison de modifications manuelles effectuées via la AWS Management Console CLI ou le SDK. CloudFormation les [ensembles de modifications sensibles à la dérive](#) comparent la configuration de votre pile traitée à l'état réel des ressources, et l'IDE met en évidence ces différences afin que vous puissiez remettre les ressources en conformité avant le déploiement.

Afficher les événements de la pile

Lorsque le déploiement démarre, vous pouvez suivre la progression en temps réel depuis le CloudFormation panneau. Sous Stack Events, vous pouvez voir une liste des opérations effectuées pendant le déploiement. Chaque événement inclut des détails tels que :

- Horodatage : heure à laquelle l'événement s'est produit
- Ressource : AWS ressource spécifique créée, mise à jour ou supprimée
- État : état actuel de l'opération (par exemple CREATE_IN_PROGRESS, UPDATE_COMPLETE, ou ROLLBACK_IN_PROGRESS)
- Motif : Contexte ou messages d'erreur supplémentaires, le cas échéant

Vous pouvez également consulter les ressources et les résultats de la pile à partir de ce panneau. La vue Stack Events vous permet de suivre la progression du déploiement, d'identifier les problèmes potentiels et de confirmer si votre stack est terminé avec succès.

Initialisation d'un CloudFormation projet dans l'IDE (VS Code uniquement)

L'initialisation d'un CloudFormation projet dans l'IDE vous permet de configurer un espace de travail structuré avec les dossiers, la configuration de l'environnement et les AWS informations d'identification appropriés afin que vous puissiez valider et déployer vos modèles de manière fiable. Vous pouvez initialiser un nouveau CloudFormation projet directement depuis l'IDE pour créer cette configuration recommandée. Cette fonctionnalité est actuellement disponible dans VS Code uniquement et n'est pas prise en charge dans JetBrains IDEs.

Pour initialiser un CloudFormation projet, procédez comme suit :

- Ouvrez la palette de commandes
 - Depuis VS Code, ouvrez la palette de commandes (Ctrl+Shift+P ou Cmd+Shift+P activez-la macOS).
 - Choisissez AWS CloudFormation: CFN Init : Initialiser le projet.
- Choisissez un répertoire de projets

- Par défaut, l'IDE utilise votre répertoire de travail actuel.
- Vous pouvez modifier ce chemin vers n'importe quel dossier dans lequel vous souhaitez stocker vos CloudFormation modèles.
- Sélectionnez votre profil AWS d'identification
 - Vous êtes invité à choisir un profil AWS d'identification. Le profil sélectionné est utilisé pour la détection de l'environnement, les validations et les déploiements.
- Configurez votre environnement
 - Vous êtes invité à créer ou à sélectionner un environnement.
 - Les environnements définissent où et comment vos modèles sont déployés ou validés (par exemple, en phase de développement, de version bêta ou de production). Vous pouvez utiliser AWS CloudFormation: CFN Init : Ajouter un environnement pour sélectionner ou modifier votre environnement.
 - Vous pouvez utiliser AWS CloudFormation: CFN Init : Remove Environment pour supprimer l'environnement que vous avez sélectionné.
- (Facultatif) Importer des fichiers de paramètres
 - Si vous avez déjà des fichiers de paramètres existants, l'IDE vous permet de les importer lors de l'initialisation.
 - L'IDE détecte automatiquement les fichiers compatibles et les lie à votre projet pour les utiliser dans le cadre de la validation et du déploiement des modèles.
- Nommer et finaliser le projet
 - Entrez un nom de projet, tel que l'environnement bêta, et terminez la configuration.
 - L'IDE crée pour vous la structure initiale du projet et le fichier de configuration.

Vous pouvez exécuter des validations, prévisualiser des déploiements ou passer d'un environnement à un autre directement depuis l'IDE.

Open source

Le serveur de AWS CloudFormation langue est open source dans le cadre de la Apache-2.0 licence, offrant aux clients une transparence totale sur la manière dont les diagnostics des modèles, la validation des schémas et les analyses statiques sont effectués. Cela réduit les difficultés liées à la sécurité et à la conformité pour les clients qui ont besoin d'une visibilité au niveau de la source avant d'adopter un outillage.

La base de code est accessible au public sur GitHub : <https://github.com/aws-cloudformation/cloudformation-languageserver/>.

Vous avez besoin d'aide ?

Essayez la [CloudFormationcommunauté](#) AWS re:Post.

Générez des modèles à partir de ressources existantes avec le générateur laC

Avec l' CloudFormation infrastructure en tant que générateur de code (générateur laC), vous pouvez générer un modèle à l'aide de AWS ressources provisionnées dans votre compte qui ne sont pas déjà gérées par CloudFormation.

Voici les avantages du générateur laC :

- Gérez des applications CloudFormation entières ou migrez-les vers une AWS CDK application.
- Générez des modèles sans avoir à décrire une ressource propriété par propriété, puis à les traduire en syntaxe JSON ou YAML.
- Utilisez le modèle pour répliquer les ressources d'un nouveau compte ou d'une nouvelle région.

Le processus de génération laC comprend les étapes suivantes :

1. Analyse des ressources : la première étape consiste à lancer une analyse de vos ressources. Ce scan concerne l'ensemble de la région et expire au bout de 30 jours. Pendant ce temps, vous pouvez créer plusieurs modèles à partir du même scan.
2. Création de votre modèle — Pour créer le modèle, deux options s'offrent à vous :
 - Créez un nouveau modèle à partir de zéro et ajoutez-y les ressources numérisées et les ressources associées.
 - Utilisez une CloudFormation pile existante comme point de départ et ajoutez les ressources numérisées et les ressources associées à son modèle.
3. Importer des ressources : utilisez votre modèle pour importer les ressources sous forme de CloudFormation pile ou pour les migrer vers une AWS CDK application.

La fonctionnalité de générateur IaC est disponible dans toutes les régions commerciales et prend en charge de nombreux types de ressources AWS courants. Pour obtenir la liste complète des ressources prises en charge, consultez [Prise en charge des types de ressources](#).

Rubriques

- [Considérations](#)
- [Autorisations IAM requises pour l'analyse des ressources](#)
- [Commandes couramment utilisées pour la génération, la gestion et la suppression de modèles](#)
- [Migrer un modèle vers AWS CDK](#)
- [Lancer une analyse des ressources avec le générateur CloudFormation IaC](#)
- [Afficher le résumé du scan dans la CloudFormation console](#)
- [Créez un CloudFormation modèle à partir de ressources scannées avec le générateur IaC](#)
- [Création d'une CloudFormation pile à partir de ressources numérisées](#)
- [Résolution de propriétés en écriture seule](#)

Considérations

Vous pouvez générer des modèles JSON ou YAML pour les AWS ressources auxquelles vous avez accès en lecture. Les modèles pour la fonctionnalité du générateur IaC modélisent les ressources cloud de manière fiable et rapide sans avoir à décrire une ressource propriété par propriété.

Le tableau suivant répertorie les quotas disponibles pour la fonctionnalité de génération IaC.

Nom	Analyse complète	Analyse partielle
Nombre maximal de ressources pouvant être traitées lors d'une analyse	100 000	100 000
Nombre de scans par jour (pour les scans comportant moins de 10 000 ressources)	10	10
Nombre de scans par jour (pour les scans comportant plus de 10 000 ressources)	1	1
Nombre simultané de modèles générés par compte	5	5

Nom	Analyse complète	Analyse partielle
Nombre simultané de ressources modélisées pour une génération de modèle	5	5
Nombre total de ressources pouvant être modélisées dans un modèle	500	500
Nombre maximal de modèles générés par compte	1 000	1 000

 Important

Le générateur laC ne prend en charge que les AWS ressources prises en charge par l'API Cloud Control dans votre région. Pour de plus amples informations, veuillez consulter [Prise en charge des types de ressources](#).

Autorisations IAM requises pour l'analyse des ressources

Pour analyser les ressources avec le générateur laC, votre principal IAM (utilisateur, rôle ou groupe) doit disposer des éléments suivants :

- CloudFormation autorisations de numérisation
- Autorisations de lecture pour les AWS services cibles

La portée de l'analyse est limitée aux ressources auxquelles vous avez accès en lecture. Les autorisations manquantes n'entraîneront pas l'échec de l'analyse, mais excluront ces ressources.

Pour un exemple de politique IAM accordant des autorisations de numérisation et de gestion des modèles, voir [Autoriser toutes les opérations du générateur laC](#).

Commandes couramment utilisées pour la génération, la gestion et la suppression de modèles

Les commandes couramment utilisées pour travailler avec le générateur laC sont les suivantes :

- [start-resource-scan](#) pour lancer une analyse des ressources du compte dans un Région AWS.

- [describe-resource-scan](#) pour suivre la progression d'une analyse des ressources.
- [list-resource-scans](#) pour répertorier les analyses de ressources dans un Région AWS.
- [list-resource-scan-resources](#) pour répertorier les ressources trouvées lors de l'analyse des ressources.
- [list-resource-scan-related-resources](#) pour répertorier les ressources liées à vos ressources numérisées.
- [create-generated-template](#) pour générer un CloudFormation modèle à partir d'un ensemble de ressources numérisées.
- [update-generated-template](#) pour mettre à jour le modèle généré.
- [describe-generated-template](#) pour renvoyer des informations sur un modèle généré.
- [list-generated-templates](#) pour répertorier tous les modèles générés dans votre compte et dans votre région actuelle.
- [delete-generated-template](#) pour supprimer un modèle généré.

Migrer un modèle vers AWS CDK

AWS Cloud Development Kit (AWS CDK) Il s'agit d'un framework de développement de logiciels open source que vous pouvez utiliser pour développer, gérer et déployer des CloudFormation ressources à l'aide de langages de programmation courants.

La AWS CDK CLI fournit une intégration avec le générateur IaC. Utilisez la `cdk migrate` commande AWS CDK CLI pour convertir le CloudFormation modèle et créer une nouvelle application CDK contenant vos ressources. Vous pouvez ensuite utiliser le AWS CDK pour gérer vos ressources et les déployer sur CloudFormation.

Pour plus d'informations, consultez la section [Migrer vers AWS CDK](#) dans le Guide du développeur AWS Cloud Development Kit (AWS CDK) .

Lancer une analyse des ressources avec le générateur CloudFormation IaC

Avant de créer un modèle à partir de ressources existantes, vous devez d'abord lancer une analyse des ressources afin d'identifier vos ressources actuelles et leurs relations.

Vous pouvez démarrer une analyse des ressources à l'aide de l'une des solutions suivantes. Pour les nouveaux utilisateurs du générateur IaC, nous recommandons la première option.

- Analyser toutes les ressources (analyse complète) : analyse toutes les ressources existantes du compte courant et de la région. Ce processus d'analyse peut prendre jusqu'à 10 minutes pour 1 000 ressources.
- Analyser des ressources spécifiques (analyse partielle) : sélectionnez manuellement les types de ressources à analyser dans le compte courant et dans la région. Cette option permet un processus de numérisation plus rapide et plus ciblé, ce qui la rend idéale pour le développement itératif de modèles.

Une fois l'analyse terminée, vous pouvez choisir les ressources et leurs ressources associées à inclure lors de la génération de votre modèle. Lorsque vous utilisez l'analyse partielle, les ressources associées ne seront disponibles pendant la génération du modèle que si :

- Vous les avez spécifiquement sélectionnés avant de commencer le scan, ou
- Ils étaient tenus de découvrir les types de ressources que vous avez sélectionnés.

Par exemple, si vous sélectionnez `AWS::EKS::Nodegroup` sans sélectionner `AWS::EKS::Cluster`, le générateur IaC inclut automatiquement les ressources `AWS::EKS::Cluster` dans l'analyse, car la découverte du groupe de nœuds nécessite d'abord la découverte du cluster. Dans tous les autres cas, l'analyse n'inclura que les ressources que vous avez spécifiquement sélectionnées.

Note

Avant de continuer, vérifiez que vous disposez des autorisations nécessaires pour utiliser le générateur IaC. Pour de plus amples informations, veuillez consulter [Autorisations IAM requises pour l'analyse des ressources](#).

Rubriques

- [Lancer une analyse des ressources \(console\)](#)
- [Lancer une analyse des ressources \(AWS CLI\)](#)

Lancer une analyse des ressources (console)

Pour démarrer une analyse de tous les types de ressources (analyse complète)

1. Ouvrez la [page du générateur iAc](#) de la CloudFormation console.
2. Dans la barre de navigation située en haut de l'écran, choisissez la Région AWS qui contient les ressources à numériser.
3. Dans le panneau Analyses, choisissez Démarrer une nouvelle analyse, puis sélectionnez Analyser toutes les ressources.

Pour démarrer une analyse des ressources de types de ressources spécifiques (analyse partielle)

1. Ouvrez la [page du générateur iAc](#) de la CloudFormation console.
2. Dans la barre de navigation située en haut de l'écran, choisissez la Région AWS qui contient les ressources à numériser.
3. Dans le panneau Analyses, choisissez Démarrer une nouvelle analyse, puis sélectionnez Analyser les ressources spécifiques.
4. Dans la boîte de dialogue Lancer le scan partiel, sélectionnez jusqu'à 100 types de ressources, puis choisissez Lancer le scan.

Lancer une analyse des ressources (AWS CLI)

Pour démarrer une analyse de tous les types de ressources (analyse complète)

Utilisez la commande [start-resource-scan](#) suivante. *us-east-1* Remplacez-le par Région AWS celui qui contient les ressources à analyser.

```
aws cloudformation start-resource-scan --region us-east-1
```

Si elle aboutit, cette commande renvoie l'ARN de l'analyse. Notez l'ARN dans la propriété ResourceScanId. Vous en avez besoin pour créer votre modèle.

```
{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60"
}
```

Pour démarrer une analyse des ressources de types de ressources spécifiques (analyse partielle)

1. Utilisez la commande [cat](#) suivante pour stocker les types de ressources que vous souhaitez analyser dans un fichier JSON nommé `config.json` dans votre répertoire personnel. Voici un exemple de configuration de scan qui recherche les instances Amazon EC2, les groupes de sécurité et toutes les ressources Amazon S3.

```
$ cat > config.json
[
  {
    "Types": [
      "AWS::EC2::Instance",
      "AWS::EC2::SecurityGroup",
      "AWS::S3::*"
    ]
  }
]
```

2. Utilisez la [start-resource-scan](#) commande avec l'--scan-filters option, ainsi que le `config.json` fichier que vous avez créé, pour démarrer l'analyse partielle. `us-east-1` Remplacez-le par Région AWS celui qui contient les ressources à analyser.

```
aws cloudformation start-resource-scan --scan-filters file://config.json --
region us-east-1
```

Si elle aboutit, cette commande renvoie l'ARN de l'analyse. Notez l'ARN dans la propriété `ResourceScanId`. Vous en avez besoin pour créer votre modèle.

```
{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-
a3ef-3e6ecfa5da60"
}
```

Pour surveiller la progression de l'analyse des ressources

Utilisez la commande [describe-resource-scan](#). Pour l'option --resource-scan-id, remplacez l'exemple d'ARN par l'ARN réel.

```
aws cloudformation describe-resource-scan --region us-east-1 \
```



```
--resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60
```

Si elle aboutit, cette commande renvoie une sortie similaire à ce qui suit :

```
{
  "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
  "Status": "COMPLETE",
  "StartTime": "2023-08-21T03:10:38.485000+00:00",
  "EndTime": "2023-08-21T03:20:28.485000+00:00",
  "PercentageCompleted": 100.0,
  "ResourceTypes": [
    "AWS::CloudFront::CachePolicy",
    "AWS::CloudFront::OriginRequestPolicy",
    "AWS::EC2::DHCPOptions",
    "AWS::EC2::InternetGateway",
    "AWS::EC2::KeyPair",
    "AWS::EC2::NetworkAcl",
    "AWS::EC2::NetworkInsightsPath",
    "AWS::EC2::NetworkInterface",
    "AWS::EC2::PlacementGroup",
    "AWS::EC2::Route",
    "AWS::EC2::RouteTable",
    "AWS::EC2::SecurityGroup",
    "AWS::EC2::Subnet",
    "AWS::EC2::SubnetCidrBlock",
    "AWS::EC2::SubnetNetworkAclAssociation",
    "AWS::EC2::SubnetRouteTableAssociation",
    ...
  ],
  "ResourcesRead": 676
}
```

Pour une analyse partielle, le résultat sera similaire à ce qui suit :

```
{
  "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
  "Status": "COMPLETE",
  "StartTime": "2025-03-06T18:24:19.542000+00:00",
  "EndTime": "2025-03-06T18:25:23.142000+00:00",
  "PercentageCompleted": 100.0,
```

```
"ResourceTypes": [
    "AWS::EC2::Instance",
    "AWS::EC2::SecurityGroup",
    "AWS::S3::Bucket",
    "AWS::S3::BucketPolicy"
],
"ResourcesRead": 65,
"ScanFilters": [
    {
        "Types": [
            "AWS::EC2::Instance",
            "AWS::EC2::SecurityGroup",
            "AWS::S3:*"
        ]
    }
]
```

Pour une description des champs de la sortie, reportez-vous [DescribeResourceScan](#) à la référence de l'AWS CloudFormation API.

Afficher le résumé du scan dans la CloudFormation console

Une fois l'analyse terminée, vous pouvez visualiser les ressources détectées pendant l'analyse afin d'identifier la concentration des ressources entre les différents types de produits.

Pour afficher des informations sur les ressources trouvées lors de l'analyse

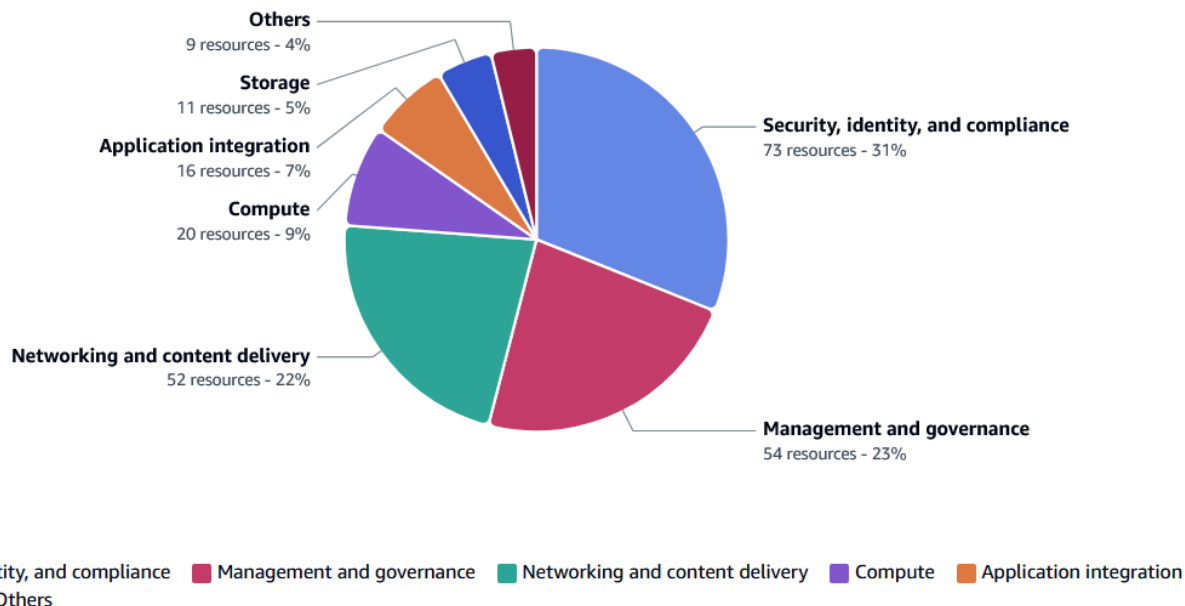
1. Ouvrez la [page du générateur iAC](#) de la CloudFormation console.
2. Dans la barre de navigation en haut de l'écran, choisissez Région AWS celui qui contient l'analyse des ressources à afficher.
3. Depuis le volet de navigation, choisissez Générateur IaC.
4. Sous Répartition des ressources numérisées, vous trouverez une ventilation visuelle des ressources numérisées par type de produit, par exemple, calcul et stockage.
5. Pour personnaliser le nombre de types de produits affichés, choisissez Filtrer les données affichées. Cela vous permet d'adapter la visualisation afin de vous concentrer sur les types de produits qui vous intéressent le plus.
6. Sur le côté droit de la page se trouve le panneau des détails du résumé du scan. Pour ouvrir le panneau, sélectionnez l'icône ouverture du panneau.

Scanned resources breakdown 235

View a visual breakdown of your scanned resources by product types. Select a product type for a further breakdown.

Filter displayed data

Filter by product type ▼



Créez un CloudFormation modèle à partir de ressources scannées avec le générateur IaC

Cette rubrique explique comment créer des règles à l'aide de la fonctionnalité IaC Generator.

Création d'un modèle à partir de ressources numérisées (console)

Pour créer un modèle de pile à partir de ressources numérisées

1. Ouvrez la [page du générateur iAC](#) de la CloudFormation console.
2. Dans la barre de navigation située en haut de l'écran, choisissez la Région AWS qui contient les ressources numérisées.
3. Dans la section Modèles, choisissez Créer un modèle.
4. Choisissez Démarrer à partir d'un nouveau modèle.
 - a. Dans Nom du modèle, saisissez un nom pour le modèle.

- b. (Facultatif) Configurez votre Politique de suppression et Mettez à jour la politique de remplacement.
 - c. Choisissez Next pour ajouter des ressources numérisées au modèle.
5. Pour Ajouter des ressources numérisées, parcourez la liste des ressources numérisées et sélectionnez les ressources que vous souhaitez ajouter à votre modèle. Vous pouvez filtrer les ressources par identifiant de ressource, type de ressource ou balises. Les filtres sont mutuellement inclusifs.
6. Une fois que vous avez ajouté toutes les ressources nécessaires à votre modèle, sélectionnez Suivant pour quitter la page Ajouter des ressources numérisées et passer à la page Ajouter des ressources associées.
7. Consultez une liste recommandée de ressources connexes. Les ressources associées, telles que les instances Amazon EC2 et les groupes de sécurité, sont interdépendantes et appartiennent généralement à la même charge de travail. Sélectionnez les ressources connexes que vous souhaitez inclure dans le modèle généré.

 Note

Nous vous suggérons d'ajouter toutes les ressources associées à ce modèle.

8. Passez en revue les détails du modèle, les ressources numérisées et les ressources associées.
9. Choisissez Créer un modèle pour quitter la page Révision et création et créer le modèle.

Création d'un modèle à partir de ressources numérisées (AWS CLI)

Pour créer un modèle de pile à partir de ressources numérisées

1. Utilisez la [list-resource-scan-resources](#) commande pour répertorier les ressources trouvées lors de l'analyse, en spécifiant éventuellement l'`--resource-identifier` option permettant de limiter le résultat. Pour l'option `--resource-scan-id`, remplacez l'exemple d'ARN par l'ARN réel.

```
aws cloudformation list-resource-scan-resources \
  --resource-scan-id arn:aws:cloudformation:us-
east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resource-identifier MyApp
```

Voici un exemple de réponse, qui `ManagedByStack` indique si CloudFormation la ressource est déjà gérée. Copiez la sortie. Vous en aurez besoin à l'étape suivante.

```
{
  "Resources": [
    {
      "ResourceType": "AWS::EKS::Cluster",
      "ResourceIdentifier": {
        "ClusterName": "MyAppClusterName"
      },
      "ManagedByStack": false
    },
    {
      "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
      "ResourceIdentifier": {
        "AutoScalingGroupName": "MyAppASGName"
      },
      "ManagedByStack": false
    }
  ]
}
```

Pour une description des champs de la sortie, reportez-vous [ScannedResource](#) à la référence de l'AWS CloudFormation API.

2. Utilisez la commande `cat` pour stocker les types de ressources et les identifiants dans un fichier JSON nommé `resources.json` dans votre répertoire personnel. Voici un exemple de JSON basé sur l'exemple de sortie de l'étape précédente.

```
$ cat > resources.json
[
  {
    "ResourceType": "AWS::EKS::Cluster",
    "ResourceIdentifier": {
      "ClusterName": "MyAppClusterName"
    }
  },
  {
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
    "ResourceIdentifier": {
      "AutoScalingGroupName": "MyAppASGName"
    }
  }
]
```

```
}
]
```

3. Utilisez la commande [list-resource-scan-related-resources](#), ainsi que le `resources.json` fichier que vous avez créé, pour répertorier les ressources associées à vos ressources numérisées.

```
aws cloudformation list-resource-scan-related-resources \
  --resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resources file://resources.json
```

Voici un exemple de réponse, qui `ManagedByStack` indique si CloudFormation la ressource est déjà gérée. Ajoutez ces ressources au fichier JSON créé à l'étape précédente. Vous aurez besoin de créer votre modèle.

```
{
  "RelatedResources": [
    {
      "ResourceType": "AWS::EKS::Nodegroup",
      "ResourceIdentifier": {
        "NodegroupName": "MyAppNodegroupName"
      },
      "ManagedByStack": false
    },
    {
      "ResourceType": "AWS::IAM::Role",
      "ResourceIdentifier": {
        "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
      },
      "ManagedByStack": false
    }
  ]
}
```

Pour une description des champs de la sortie, reportez-vous [ScannedResource](#) à la référence de l'AWS CloudFormation API.

 Note

La liste de ressources en entrée ne peut pas dépasser une longueur de 100. Pour répertorier les ressources associées pour plus de 100 ressources, exécutez la commande `list-resource-scan-related-resources` par lots de 100 et consolidez les résultats.

Sachez que la sortie peut contenir des ressources dupliquées dans la liste.

4. Utilisez la [create-generated-template](#) commande pour créer un nouveau modèle de pile, comme suit, avec les modifications suivantes :

- `us-east-1` Remplacez-le par Région AWS celui qui contient les ressources numérisées.
- Remplacez `MyTemplate` par le nom du modèle à créer.

```
aws cloudformation create-generated-template --region us-east-1 \  
--generated-template-name MyTemplate \  
--resources file://resources.json
```

Voici un exemple de fichier `resources.json`.

```
[  
  {  
    "ResourceType": "AWS::EKS::Cluster",  
    "LogicalResourceId": "MyCluster",  
    "ResourceIdentifier": {  
      "ClusterName": "MyAppClusterName"  
    }  
  },  
  {  
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",  
    "LogicalResourceId": "MyASG",  
    "ResourceIdentifier": {  
      "AutoScalingGroupName": "MyAppASGName"  
    }  
  },  
  {  
    "ResourceType": "AWS::EKS::Nodegroup",  
    "LogicalResourceId": "MyNodegroup",  
    "ResourceIdentifier": {
```

```

        "NodegroupName": "MyAppNodegroupName"
      },
      {
        "ResourceType": "AWS::IAM::Role",
        "LogicalResourceId": "MyRole",
        "ResourceIdentifier": {
          "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
        }
      }
    ]
  }
}

```

Si elle réussit, cette commande renvoie ce qui suit.

```

{
  "Arn":
    "arn:aws:cloudformation:region:account-id:generatedtemplate/7fc8512c-d8cb-4e02-
    b266-d39c48344e48",
  "Name": "MyTemplate"
}

```

Création d'une CloudFormation pile à partir de ressources numérisées

Après avoir créé votre modèle, vous pouvez prévisualiser le modèle généré avec Infrastructure Composer avant de créer la pile et d'importer les ressources numérisées. Cela vous permet de visualiser l'architecture complète de l'application avec les ressources et leurs relations. Pour plus d'informations sur Infrastructure Composer, reportez-vous à la section [Créez des modèles visuellement avec Infrastructure Composer](#).

Pour créer la pile et importer les ressources numérisées

1. Ouvrez la [page du générateur iAC](#) de la CloudFormation console.
2. Dans la barre de navigation en haut de l'écran, choisissez le modèle correspondant Région AWS à votre modèle.
3. Cliquez sur l'onglet Modèles, puis choisissez le nom de votre modèle pour afficher plus d'informations.
4. Dans l'onglet Définition du modèle, en haut de la section Modèle, vous pouvez passer du modèle de la syntaxe YAML à la syntaxe JSON selon vos préférences.

5. Vérifiez les détails de votre modèle pour vous assurer que tout est correctement configuré. Pour faciliter la consultation et la compréhension du modèle, vous pouvez passer de l'affichage du code par défaut à une vue graphique de l'infrastructure décrite dans le modèle à l'aide d'Infrastructure Composer. Pour ce faire, sous Modèle, choisissez Canvas au lieu de Modèle.

Actions du canvas

- Pour vous concentrer sur les détails d'une ressource spécifique dans votre modèle, double-cliquez sur une carte pour afficher le panneau Propriétés de la ressource.
 - Pour disposer et organiser visuellement les cartes sur le canevas, choisissez Arrangement dans le coin supérieur gauche du canevas.
 - Pour zoomer et dézoomer sur votre toile, utilisez les commandes de zoom situées dans le coin inférieur droit de la toile.
6. Pour afficher une ressource spécifique dans la console, sélectionnez l'onglet Ressources de modèle, puis sélectionnez l'ID physique de la ressource que vous souhaitez consulter. Cela vous amène à la console de cette ressource spécifique. Vous pouvez également ajouter, supprimer et resynchroniser des ressources dans la définition du modèle à partir de l'onglet Ressources du modèle.
 7. Dans l'onglet Définition du modèle, le générateur IaC peut émettre des avertissements concernant les ressources qui contiennent des propriétés en écriture seule. Après avoir pris connaissance des avertissements, vous pouvez télécharger le modèle généré et y apporter les modifications nécessaires. Pour de plus amples informations, veuillez consulter [Résolution de propriétés en écriture seule](#).
 8. Lorsque vous êtes satisfait de la définition de votre modèle, dans l'onglet Définition du modèle, sélectionnez Importer dans la pile, puis cliquez sur Suivant.
 9. Dans le panneau Spécifier la pile de la page Spécifier les détails de la pile, entrez le nom de votre pile, puis sélectionnez Suivant.
 10. Vérifiez et entrez les paramètres de la pile. Choisissez Suivant.
 11. Vérifiez vos options sur la page Vérifier les modifications et sélectionnez Suivant.
 12. Vérifiez vos informations sur la page Révision et importation, puis choisissez Importer des ressources.

Résolution de propriétés en écriture seule

Avec le générateur CloudFormation iAC, vous pouvez générer un modèle à l'aide des ressources fournies dans votre compte qui ne sont pas déjà gérées par CloudFormation. Cependant, certaines propriétés de ressources sont désignées comme étant en écriture seule, ce qui signifie qu'elles peuvent être écrites mais ne peuvent pas être lues CloudFormation, par exemple par un mot de passe de base de données.

Lorsque vous générez CloudFormation des modèles à partir de ressources existantes, les propriétés en écriture seule constituent un défi. Dans la plupart des cas, CloudFormation convertit ces propriétés en paramètres dans le modèle généré. Cela vous permet de saisir les propriétés sous forme de valeurs de paramètres lors des opérations d'importation. Cependant, il existe des scénarios dans lesquels cette conversion n'est pas possible et CloudFormation gère ces cas différemment.

Propriétés mutuellement exclusives

Certaines ressources possèdent plusieurs ensembles de propriétés qui s'excluent mutuellement, dont au moins certaines sont en écriture seule. Dans ces cas, le générateur laC ne peut pas déterminer quel ensemble de propriétés exclusives a été appliqué à la ressource lors de sa création. Par exemple, vous pouvez fournir le code pour un [AWS::Lambda::Function](#) qui utilise l'un de ces ensembles de propriétés.

- Code/S3Bucket, Code/S3Key, et éventuellement Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

Toutes ces propriétés sont accessibles en écriture seule. Le générateur laC sélectionne l'un des ensembles de propriétés exclusifs et les ajoute au modèle généré. Des paramètres sont ajoutés pour chacune des propriétés en écriture seule. Les noms des paramètres incluent OneOf et leurs descriptions indiquent que la propriété correspondante peut être remplacée par d'autres propriétés exclusives. Le générateur laC définit un type d'avertissement `MUTUALLY_EXCLUSIVE_PROPERTIES` pour les propriétés incluses.

Types mutuellement exclusifs

Dans certains cas, une propriété en écriture seule peut comporter plusieurs types de données. Par exemple, la propriété Body de [AWS::ApiGateway::RestApi](#) peut être un objet ou un string. Dans

ce cas, le générateur IaC inclut la propriété dans le modèle généré en utilisant le type de `string` et définit un type d'avertissement `MUTUALLY_EXCLUSIVE_TYPES`.

Propriétés de Array

Si une propriété en écriture seule est de type `array`, le générateur IaC ne peut pas l'inclure dans le modèle généré, car les paramètres ne peuvent être que des valeurs scalaires. Dans ce cas, la propriété est omise du modèle et le type d'avertissement `UNSUPPORTED_PROPERTIES` est défini.

Propriétés facultatives

Pour les propriétés en écriture seule facultatives, le générateur IaC ne peut pas détecter si la propriété a été utilisée lors de la configuration de la ressource. Dans ce cas, la propriété est omise du modèle généré et le type d'avertissement `UNSUPPORTED_PROPERTIES` est défini.

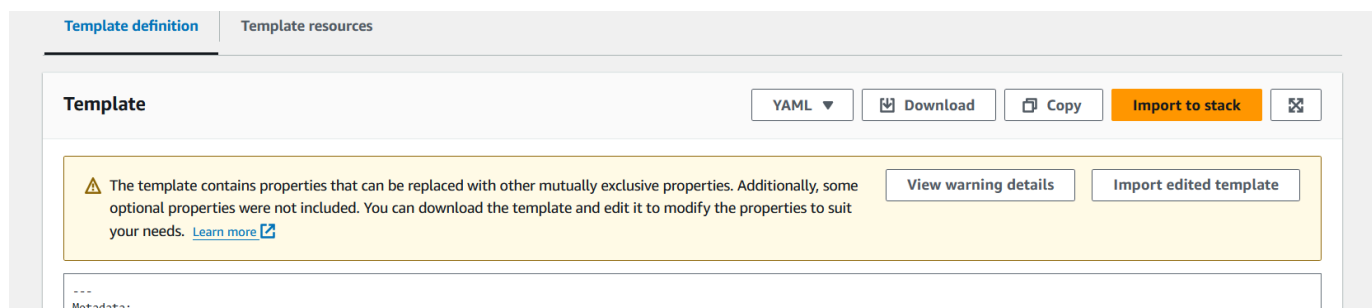
Avertissements et prochaines étapes

Pour déterminer quelles propriétés sont en écriture seule, vous devez consulter les avertissements renvoyés par la console du générateur IaC. La [référence aux types de ressources et de propriétés AWS](#) n'indique pas si une propriété est en écriture seule ou si elle prend en charge plusieurs types.

Vous pouvez également voir quelles propriétés sont en écriture seule à partir des schémas du fournisseur de ressources. Pour télécharger les schémas du fournisseur de ressources, consultez les schémas du [fournisseur de CloudFormation ressources](#).

Pour résoudre les problèmes liés aux propriétés en écriture seule

1. Ouvrez la [page du générateur iAC](#) de la CloudFormation console.
2. Dans la barre de navigation en haut de l'écran, choisissez le modèle correspondant Région AWS à votre modèle.
3. Sélectionnez l'onglet Modèles, puis choisissez le nom du modèle que vous avez créé.
4. Dans l'onglet Définition du modèle, lorsque le modèle généré comprend des ressources avec des propriétés en écriture seule, la console du générateur IaC affiche un avertissement avec un résumé du type de problèmes. Par exemple :



5. Sélectionnez Afficher les détails de l'avertissement pour plus d'informations. Les ressources avec des propriétés en écriture seule sont identifiées par l'ID logique utilisé dans le modèle généré et le type de ressource.

Utilisez la liste des avertissements pour identifier les ressources avec des propriétés en écriture seule et examinez chaque ressource afin de déterminer les modifications (le cas échéant) à apporter au modèle généré.

Resource warnings (1)
×

▼ LambdaFunction001aClambda002fd6K (AWS::Lambda::Function) [Learn more](#)

Unsupported properties	Mutually exclusive properties
1. SnapStart/ApplyOn Applying SnapStart setting on function resource type. Possible values: [PublishedVersions, None] This property can be replaced with other types	1. Code/S3Bucket An Amazon S3 bucket in the same AWS Region as your function. The bucket can be in a different AWS account. This property can be replaced with other exclusive properties
2. Code/S3ObjectVersion For versioned objects, the version of the deployment package object to use. This property can be replaced with other exclusive properties	2. Code/S3Key The Amazon S3 key of the deployment package. This property can be replaced with other exclusive properties

6. Si votre modèle doit être mis à jour pour résoudre des problèmes liés aux propriétés en écriture seule, procédez comme suit :
 - a. Sélectionnez Télécharger pour télécharger une copie du modèle.
 - b. Modifiez votre modèle.
 - c. Une fois les modifications terminées, vous pouvez sélectionner le bouton Importer le modèle modifié pour poursuivre le processus d'importation.

Comment résoudre les problèmes liés aux propriétés en écriture seule dans les ressources AWS::ApiGateway::RestAPI

Cette rubrique explique comment résoudre les problèmes liés aux propriétés en écriture seule dans les ressources [AWS::ApiGateway::RestApi](#) lorsque vous utilisez le générateur IaC.

Problème

Lorsqu'un modèle généré contient des ressources `AWS::ApiGateway::RestApi`, des avertissements sont générés indiquant que les propriétés `Body`, `BodyS3Location`, et `CloneFrom` sont identifiées comme `UNSUPPORTED_PROPERTIES`. Cela est dû au fait qu'il s'agit de propriétés facultatives en écriture seule. Le générateur IaC ne sait pas si ces propriétés ont déjà été appliquées à la ressource. Par conséquent, il omet ces propriétés dans le modèle généré.

Résolution

Pour définir la propriété `Body` de votre API REST, mettez à jour votre modèle généré.

1. Utilisez l'action API Amazon API Gateway [GetExport](#) pour télécharger l'API. Par exemple, en utilisant la [aws apigateway get-export](#) AWS CLI commande. Pour plus d'informations, consultez la section [Exporter une API REST depuis API Gateway](#) dans le Guide du développeur API Gateway.
2. Récupérez la propriété `Body` à partir de la réponse de l'action de l'API `GetExport`. Chargez-la sur un compartiment Amazon S3.
3. Téléchargez le modèle généré.
4. Ajoutez les propriétés `BodyS3Location/Bucket` et `BodyS3Location/Key` au modèle, en spécifiant le nom du compartiment et la clé où `Body` est stocké.
5. Ouvrez le modèle généré dans la console du générateur IaC et choisissez Importer le modèle modifié.

Comment résoudre les problèmes liés aux propriétés en écriture seule dans les ressources `AWS::Lambda::Function`

Cette rubrique explique comment résoudre les problèmes liés aux propriétés en écriture seule dans les ressources [AWS::Lambda::Function](#) lorsque vous utilisez le générateur IaC.

Problème

La ressource `AWS::Lambda::Function` dispose de trois ensembles de propriétés mutuellement exclusifs pour spécifier le code Lambda :

- Les propriétés `Code/S3Bucket` et `Code/S3Key`, et éventuellement la propriété `Code/S3ObjectVersion`

- propriété `Code/ImageUri`
- propriété `Code/ZipFile`

Un seul de ces ensembles peut être utilisé pour une ressource `AWS::Lambda::Function` donnée.

Le générateur IaC ne peut pas déterminer quel ensemble de propriétés exclusives en écriture seule a été utilisé pour créer ou mettre à jour la ressource. Par conséquent, il n'inclut que le premier ensemble de propriétés dans le modèle généré. Les propriétés `Code/ImageUri` et `Code/ZipFile` sont omises.

En outre, le générateur IaC émet les avertissements suivants :

- **MUTUALLY_EXCLUSIVE_PROPERTIES** : avertit que `Code/S3Bucket` et `Code/S3Key` sont identifiées comme des propriétés mutuellement exclusives.
- **UNSUPPORTED_PROPERTIES** : avertit que la propriété `Code/S3ObjectVersion` n'est pas prise en charge.

Pour inclure des ressources `AWS::Lambda::Function` dans un modèle généré, vous devez télécharger et mettre à jour le modèle avec les propriétés de code correctes.

Résolution

Si vous stockez votre code Lambda dans un compartiment Amazon S3 et n'utilisez pas la propriété **`S3ObjectVersion`**, vous pouvez importer le modèle généré sans aucune modification. Le générateur IaC vous demandera le compartiment Amazon S3 et la clé en tant que paramètres de modèle pendant l'opération d'importation.

Si vous stockez votre code Lambda dans un référentiel Amazon ECR, vous pouvez mettre à jour votre modèle en suivant les instructions suivantes :

1. Téléchargez le modèle généré.
2. Supprimez les propriétés et les paramètres correspondants pour les propriétés `Code/S3Bucket` et `Code/S3Key` du modèle généré.
3. Remplacez les propriétés supprimées dans le modèle généré par la propriété `Code/ImageUri`, en spécifiant l'URL du référentiel Amazon ECR.
4. Ouvrez le modèle généré dans la console du générateur IAC et cliquez sur le bouton Importer le modèle modifié.

Si vous stockez votre code Lambda dans un fichier zip, vous pouvez mettre à jour votre modèle en suivant les instructions suivantes :

1. Téléchargez le modèle généré.
2. Supprimez les propriétés et les paramètres correspondants pour les propriétés Code/S3Bucket et Code/S3Key du modèle généré.
3. Remplacez les propriétés supprimées dans le modèle généré par la propriété Code/ZipFile.
4. Ouvrez le modèle généré dans la console du générateur IAC et cliquez sur le bouton Importer le modèle modifié.

Si vous ne disposez pas d'une copie de votre code Lambda, vous pouvez mettre à jour votre modèle en suivant les instructions suivantes :

1. Utilisez l'action AWS Lambda [GetFunction](#)API (par exemple, à l'aide de la [aws lambda get-function](#) AWS CLI commande).
2. Dans la réponse, le paramètre RepositoryType indique S3 si le code se trouve dans un compartiment Amazon S3 ou ECR s'il se trouve dans un référentiel Amazon ECR.
3. Dans la réponse, le paramètre Location contient une URL pré-signée que vous pouvez utiliser pour télécharger le package de déploiement pendant 10 minutes. Téléchargez le code.
4. Téléchargez le code dans un compartiment Amazon S3.
5. Exécutez une opération d'importation avec le modèle généré et fournissez le nom du compartiment et la clé comme valeurs de paramètre.

Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques

Les références dynamiques vous permettent de spécifier facilement des valeurs externes stockées et gérées dans d'autres services et de dissocier les informations sensibles de vos infrastructure-as-code modèles. CloudFormation récupère la valeur de la référence spécifiée lorsque cela est nécessaire lors des opérations de pile et de modification des ensembles.

Grâce aux références dynamiques, vous pouvez :

- Utiliser des chaînes sécurisées : pour les données sensibles, utilisez toujours des paramètres de chaîne sécurisés dans AWS Systems Manager Parameter Store ou des secrets AWS Secrets Manager pour vous assurer que vos données sont chiffrées au repos.
- Limiter l'accès : limitez l'accès aux paramètres du magasin de paramètres ou aux secrets du gestionnaire de secrets aux seuls principaux et rôles autorisés.
- Rotation des informations d'identification : effectuez régulièrement la rotation des données sensibles stockées dans Parameter Store ou Secrets Manager afin de maintenir un niveau de sécurité élevé.
- Rotation automatique : tirez parti des fonctionnalités de rotation automatique de Secrets Manager pour mettre à jour et distribuer périodiquement vos données sensibles dans vos applications et environnements.

Considérations d'ordre général

Voici les considérations générales à prendre en compte avant de spécifier des références dynamiques dans vos CloudFormation modèles :

- Évitez d'inclure des références dynamiques ou des données sensibles dans les propriétés des ressources qui font partie de l'identifiant principal d'une ressource. CloudFormation peut utiliser la valeur réelle en texte brut dans l'identifiant de ressource principal, ce qui peut présenter un risque de sécurité. Cet ID de ressource peut apparaître dans toutes les sorties ou destinations dérivées.

Pour déterminer quelles propriétés de ressource constituent l'identifiant principal d'un type de ressource, reportez-vous à la documentation de référence de cette ressource dans la référence des [types de ressources et de propriétés AWS](#). Dans la section Valeurs renvoyées, la valeur de retour de la fonction Ref représente les propriétés de la ressource qui constituent l'identifiant principal du type de ressource.

- Vous pouvez inclure jusqu'à 60 références dynamiques dans un modèle de pile.
- Si vous utilisez des transformations (comme `AWS::Include` ou `AWS::Serverless`), CloudFormation cela ne résout pas les références dynamiques avant d'appliquer la transformation. Au lieu de cela, il transmet la chaîne littérale de la référence dynamique à la transformation et résout les références lorsque vous exécutez le jeu de modifications à l'aide du modèle.
- Les références dynamiques ne peuvent pas être utilisées pour les valeurs sécurisées (telles que celles stockées dans Parameter Store ou Secrets Manager) dans les ressources personnalisées.
- Les références dynamiques ne sont pas non plus prises en charge dans les métadonnées `AWS::CloudFormation::Init` et les propriétés Amazon EC2 UserData.

- Ne créez pas de référence dynamique qui se termine par une barre oblique inverse (\). CloudFormation ne peut pas résoudre ces références, ce qui entraînera l'échec des opérations de pile.

Les rubriques suivantes fournissent des informations et d'autres considérations sur l'utilisation de références dynamiques.

Rubriques

- [Obtenir une valeur en texte brut depuis le magasin de paramètres Systems Manager](#)
- [Obtenir une valeur de chaîne sécurisée depuis le magasin de paramètres Systems Manager](#)
- [Obtenir un secret ou une valeur secrète à partir du Secrets Manager](#)

Obtenir une valeur en texte brut depuis le magasin de paramètres Systems Manager

Lorsque vous créez un CloudFormation modèle, vous pouvez utiliser des valeurs en texte brut stockées dans Parameter Store. Parameter Store est une fonctionnalité de AWS Systems Manager. Pour une introduction à Parameter Store, consultez la section [Parameter Store AWS Systems Manager](#) dans le Guide de l'utilisateur AWS Systems Manager .

Pour utiliser une valeur en texte clair provenant de Parameter Store dans votre modèle, vous utilisez une référence dynamique ssm. Cette référence vous permet d'accéder aux valeurs des paramètres de type String ou StringList dans Parameter Store.

Pour vérifier quelle version d'une référence dynamique ssm sera utilisée dans une opération de pile, créez un ensemble de modifications pour l'opération de pile. Ensuite, vérifiez le modèle traité dans l'onglet Modèle. Pour de plus amples informations, veuillez consulter [Création d'un ensemble de modifications pour une CloudFormation pile](#).

Lorsque vous utilisez des références dynamiques ssm, il y a quelques points importants à garder à l'esprit :

- CloudFormation ne prend pas en charge la détection de dérive sur les références dynamiques. Pour les références dynamiques ssm pour lesquelles vous n'avez pas spécifié la version du paramètre, nous vous recommandons, si vous mettez à jour la version du paramètre dans Systems Manager, d'effectuer également une opération de mise à jour de pile sur toutes les piles qui incluent la référence dynamique ssm, afin d'extraire la dernière version du paramètre.

- Pour utiliser une référence ssm dynamique dans la `Parameters` section de votre CloudFormation modèle, vous devez inclure un numéro de version. CloudFormation ne vous permet pas de référencer une valeur du Parameter Store sans numéro de version dans cette section. Vous pouvez également définir votre paramètre comme un paramètre de type Systems Manager dans votre modèle. Dans ce cas, vous pouvez spécifier une clé de paramètre Systems Manager comme valeur par défaut pour votre paramètre. CloudFormation récupérera ensuite la dernière version de la valeur du paramètre dans Parameter Store, sans que vous ayez à spécifier de numéro de version. Cela peut rendre vos modèles plus simples et plus faciles à gérer. Pour de plus amples informations, veuillez consulter [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#).
- Pour les ressources personnalisées, CloudFormation résout les références ssm dynamiques avant d'envoyer la demande à la ressource personnalisée.
- CloudFormation ne prend pas en charge l'utilisation de références dynamiques pour référencer un paramètre partagé par un autre Compte AWS.
- CloudFormation ne prend pas en charge l'utilisation des libellés de paramètres de Systems Manager dans les références dynamiques.

Permissions

Pour spécifier un paramètre stocké dans le magasin de paramètres Systems Manager, vous devez disposer d'une autorisation pour appeler [GetParameters](#) le paramètre spécifié. Pour savoir comment créer des politiques IAM qui fournissent l'accès à des paramètres Systems Manager spécifiques, consultez la section [Restriction de l'accès aux paramètres Systems Manager à l'aide des politiques IAM](#) dans le Guide de l'utilisateur AWS Systems Manager .

Modèle de référence

Pour référencer une valeur en texte brut stockée dans le magasin de paramètres Systems Manager dans votre CloudFormation modèle, utilisez le modèle de ssm référence suivant.

```
{{resolve:ssm:parameter-name:version}}
```

Votre référence doit respecter le modèle d'expression régulière suivant pour `parameter-name` et `version` :

```
{{resolve:ssm:[a-zA-Z0-9_.\-/+](?:\d+)?}}
```

parameter-name

Nom du paramètre dans Parameter Store. Le nom du paramètre est sensible à la casse.

Obligatoire.

version

Nombre entier qui spécifie la version du paramètre à utiliser. Si vous ne spécifiez pas la version exacte, CloudFormation utilise la dernière version du paramètre chaque fois que vous créez ou mettez à jour la pile. Pour plus d'informations, consultez [Utilisation des versions de paramètre](#) dans le Guide de l'utilisateur AWS Systems Manager .

Facultatif.

Exemples

Rubriques

- [Paramètre ID d'AMI public](#)
- [Paramètre d'ID d'AMI personnalisé](#)

Paramètre ID d'AMI public

L'exemple suivant crée une instance EC2 qui fait référence à un paramètre AMI public. La référence dynamique récupère le dernier identifiant AMI Amazon Linux 2023 à partir du paramètre public. Pour plus d'informations sur les paramètres publics, consultez [Découverte des paramètres publics dans le magasin de paramètres](#) dans le Guide de l'utilisateur AWS Systems Manager .

JSON

```
{
  "Resources": {
    "MyInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
        "InstanceType": "t2.micro"
      }
    }
  }
}
```

```
}
```

YAML

```
Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
      InstanceType: t2.micro
```

Paramètre d'ID d'AMI personnalisé

L'exemple suivant crée un modèle de lancement EC2 qui fait référence à un identifiant AMI personnalisé stocké dans le magasin de paramètres. La référence dynamique récupère l'ID AMI à partir de la version **2** du paramètre *golden-ami* chaque fois qu'une instance est lancée à partir du modèle de lancement.

JSON

```
{
  "Resources": {
    "MyLaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Properties": {
        "LaunchTemplateName": {
          "Fn::Sub": "${AWS::StackName}-launch-template"
        },
        "LaunchTemplateData": {
          "ImageId": "{{resolve:ssm:golden-ami:2}}",
          "InstanceType": "t2.micro"
        }
      }
    }
  }
}
```

YAML

```
Resources:
```

```
MyLaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: '{{resolve:ssm:golden-ami:2}}'
      InstanceType: t2.micro
```

Obtenir une valeur de chaîne sécurisée depuis le magasin de paramètres Systems Manager

Dans CloudFormation, vous pouvez utiliser des données sensibles telles que des mots de passe ou des clés de licence sans les exposer directement dans vos modèles en stockant les données sensibles sous forme de « chaîne sécurisée » dans AWS Systems Manager Parameter Store. Pour une introduction à Parameter Store, consultez la section [Parameter Store AWS Systems Manager](#) dans le Guide de l'utilisateur AWS Systems Manager .

Pour utiliser une chaîne sécurisée Parameter Store dans votre modèle, vous devez utiliser une référence `ssm-secure` dynamique. CloudFormation ne stocke jamais la valeur réelle de la chaîne sécurisée. Au lieu de cela, il stocke uniquement la référence dynamique littérale, qui contient le nom du paramètre en clair de la chaîne sécurisée.

Lors de la création ou de la mise à jour de la pile, CloudFormation accède à la valeur de chaîne sécurisée selon les besoins, sans exposer la valeur réelle. Les chaînes sécurisées ne peuvent être utilisées que pour les propriétés de ressources qui prennent en charge le modèle de référence dynamique `ssm-secure`. Pour de plus amples informations, veuillez consulter [Ressources qui prennent en charge les modèles de paramètre dynamique pour les chaînes sécurisées](#).

CloudFormation ne renvoie la valeur réelle du paramètre pour les chaînes sécurisées dans aucun appel d'API. Elle renvoie uniquement la référence dynamique littérale. Lorsque vous comparez des modifications à l'aide d'ensembles de modifications, compare CloudFormation uniquement la chaîne de référence dynamique littérale. Il ne résout pas et ne compare pas les valeurs réelles des chaînes sécurisées.

Lorsque vous utilisez des références dynamiques `ssm-secure`, il y a quelques points importants à garder à l'esprit :

- CloudFormation Impossible d'accéder aux valeurs du Parameter Store à partir d'autres sources Comptes AWS.

- CloudFormation ne prend pas en charge l'utilisation d'étiquettes de paramètres ou de paramètres publics de Systems Manager dans les références dynamiques.
- Dans les régions `cn-north-1` et `cn-northwest-1`, les chaînes sécurisées ne sont pas prises en charge par Systems Manager.
- Les références dynamiques pour les valeurs sécurisées, telles que `quessm-secure`, ne sont actuellement pas prises en charge dans les ressources personnalisées.
- S'il est CloudFormation nécessaire d'annuler une mise à jour de pile et que la version précédemment spécifiée d'un paramètre de chaîne sécurisé n'est plus disponible, l'opération de restauration échouera. Dans ce cas, deux options s'offrent à vous :
 - Utilisez `CONTINUE_UPDATE_ROLLBACK` pour ignorer la ressource.
 - Recréez le paramètre de chaîne sécurisée dans Systems Manager Parameter Store et mettez-le à jour jusqu'à ce que la version du paramètre atteigne la version utilisée dans le modèle. Ensuite, utilisez `CONTINUE_UPDATE_ROLLBACK` sans ignorer la ressource.

Ressources qui prennent en charge les modèles de paramètre dynamique pour les chaînes sécurisées

Les ressources qui prennent en charge le modèle de référence dynamique `ssm-secure` comprennent :

Ressource	Type de propriété	Propriétés
<u>AWS::DirectoryService::MicrosoftANNOUNCE</u>		Password
<u>AWS::DirectoryService::SimpleANNOUNCE</u>		Password
<u>AWS::ElastiCache::ReplicationGroup</u>		AuthToken
<u>AWS::IAM::User</u>	<u>LoginProfile</u>	Password
<u>AWS::KinesisFirehose::DeliveryStream</u>	<u>RedshiftDestinationConfiguration</u>	Password
<u>AWS::OpsWorks::App</u>	<u>Source</u>	Password

Ressource	Type de propriété	Propriétés
AWS::OpsWorks::Stack	CustomCookbooksSource	Password
AWS::OpsWorks::Stack	RdsDbInstances	DbPassword
AWS::RDS::DBCluster		MasterUserPassword
AWS::RDS::DBInstance		MasterUserPassword
AWS::Redshift::Cluster		MasterUserPassword

Modèle de référence

Pour référencer une valeur de chaîne sécurisée depuis le magasin de paramètres Systems Manager dans votre CloudFormation modèle, utilisez le modèle de ssm-secure référence suivant.

```
{{resolve:ssm-secure:parameter-name:version}}
```

Votre référence doit respecter le modèle d'expression régulière suivant pour parameter-name et version :

```
{{resolve:ssm-secure:[a-zA-Z0-9_.\-/]+(:\d+)??}}
```

parameter-name

Nom du paramètre dans Parameter Store. Le nom du paramètre est sensible à la casse.

Obligatoire.

version

Nombre entier qui spécifie la version du paramètre à utiliser. Si vous ne spécifiez pas la version exacte, CloudFormation utilise la dernière version du paramètre chaque fois que vous créez ou mettez à jour la pile. Pour plus d'informations, consultez [Utilisation des versions de paramètre](#) dans le Guide de l'utilisateur AWS Systems Manager .

Facultatif.

Exemple

L'exemple suivant utilise une référence dynamique `ssm-secure` pour définir le mot de passe d'un utilisateur IAM vers une valeur sécurisée stockée dans Parameter Store. Comme indiqué, CloudFormation utilisera la version **10** du ***IAMUserPassword*** paramètre pour les opérations de pile et de modification des ensembles.

JSON

```
"MyIAMUser": {
  "Type": "AWS::IAM::User",
  "Properties": {
    "UserName": "MyUserName",
    "LoginProfile": {
      "Password": "{{resolve:ssm-secure:IAMUserPassword:10}}"
    }
  }
}
```

YAML

```
MyIAMUser:
  Type: AWS::IAM::User
  Properties:
    UserName: 'MyUserName'
    LoginProfile:
      Password: '{{resolve:ssm-secure:IAMUserPassword:10}}'
```

Obtenir un secret ou une valeur secrète à partir du Secrets Manager

Secrets Manager est un service qui vous permet de stocker et de gérer en toute sécurité des informations confidentielles telles que les identifiants de base de données, les mots de passe et les clés API tierces. Secrets Manager vous permet de stocker et de contrôler l'accès à ces secrets de manière centralisée, de sorte que vous pouvez remplacer les informations d'identification codées en dur dans votre code (y compris les mots de passe) par un appel d'API à Secrets Manager pour récupérer le secret par programmation. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur.

Pour utiliser des secrets complets ou des valeurs secrètes stockées dans Secrets Manager dans vos CloudFormation modèles, vous devez utiliser des références `secretsmanager` dynamiques.

Bonnes pratiques

Suivez ces bonnes pratiques lorsque vous utilisez les références dynamiques de Secrets Manager dans vos CloudFormation modèles :

- Utilisez des références sans version pour vos CloudFormation modèles : stockez les informations d'identification dans Secrets Manager et utilisez des références dynamiques sans spécifier `version-stage` ni `version-id` paramètres pour prendre en charge les flux de travail de rotation secrets appropriés.
- Tirez parti de la rotation automatique : utilisez la fonctionnalité de rotation automatique de Secrets Manager avec des références dynamiques sans version pour la gestion des informations d'identification. Cela garantit que vos informations d'identification sont régulièrement mises à jour sans nécessiter de modifications du modèle. Pour plus d'informations, consultez la rubrique [Rotation de secrets AWS Secrets Manager](#).
- Utilisez les références versionnées avec parcimonie : ne spécifiez des paramètres `version-stage` ou `version-id` explicites ou que dans des cas spécifiques, tels que les tests ou les situations de restauration.

Considérations

Lorsque vous utilisez des références `secretsmanager` dynamiques, il est important de garder à l'esprit les considérations suivantes :

- CloudFormation ne permet pas de savoir quelle version d'un secret a été utilisée lors des déploiements précédents. Planifiez soigneusement votre stratégie de gestion des secrets avant de mettre en œuvre des références dynamiques. Utilisez des références sans version lorsque cela est possible afin de tirer parti de la rotation automatique du secret. Surveillez et validez les mises à jour des ressources lorsque vous apportez des modifications aux configurations de références dynamiques, par exemple lors du passage de références dynamiques non versionnées à des références dynamiques versionnées, et inversement.
- La mise à jour de la seule valeur secrète dans Secrets Manager n' CloudFormation entraîne pas automatiquement la récupération de la nouvelle valeur. CloudFormation récupère la valeur secrète uniquement lors de la création de la ressource ou des mises à jour qui modifient la ressource contenant la référence dynamique.

Par exemple, supposons que votre modèle inclut une ressource [AWS::RDS::DBInstance](#) dont la propriété `MasterPassword` est définie sur une référence dynamique Secrets Manager.

Après avoir créé une pile à partir de ce modèle, vous mettez à jour la valeur du secret dans Secrets Manager. Cependant, la propriété `MasterPassword` conserve l'ancienne valeur du mot de passe.

Pour appliquer la nouvelle valeur secrète, vous devez modifier la `AWS::RDS::DBInstance` ressource dans votre CloudFormation modèle et effectuer une mise à jour de la pile.

Pour éviter ce processus manuel à l'avenir, envisagez d'utiliser Secrets Manager pour faire tourner automatiquement le secret.

- Les références dynamiques pour les valeurs sécurisées, telles que `secretsmanager`, ne sont actuellement pas prises en charge dans les ressources personnalisées.
- La référence dynamique `secretsmanager` peut être utilisée dans toutes les propriétés des ressources. L'utilisation de la référence `secretsmanager` dynamique indique que ni Secrets Manager ni CloudFormation les journaux ne doivent conserver de valeur secrète résolue. Toutefois, la valeur du secret peut s'afficher dans le service qui possède la ressource où elle est utilisée. Passez en revue vos données d'utilisation pour éviter toute fuite des données de secret.

Permissions

Pour spécifier un secret stocké dans Secrets Manager, vous devez disposer des autorisations nécessaires pour appeler le [GetSecretValue](#) du secret.

Modèle de référence

Pour référencer les secrets de Secrets Manager dans votre CloudFormation modèle, utilisez le modèle `secretsmanager` de référence suivant.

```
{{resolve:secretsmanager:secret-id:secret-string:json-key:version-stage:version-id}}
```

`secret-id`

Nom ou ARN du secret.

Pour accéder à un secret qui se trouve dans votre Compte AWS répertoire, il vous suffit de spécifier le nom du secret. Pour accéder à un secret dans un autre Compte AWS format, spécifiez l'ARN complet du secret.

Obligatoire.

secret-string

La seule valeur prise en charge est `SecretString`. La valeur par défaut est `SecretString`.

json-key

Le nom de clé de la paire clé-valeur dont vous voulez récupérer la valeur. Si vous ne spécifiez pas `json-key`, CloudFormation récupère l'intégralité du texte secret.

Ce segment peut ne pas inclure le caractère deux-points (:).

version-stage

L'étiquette intermédiaire de la version du secret à utiliser. Secrets Manager utilise des étiquettes intermédiaires pour assurer le suivi des différentes versions pendant le processus de rotation. Si vous utilisez `version-stage`, ne spécifiez pas `version-id`. Si vous ne spécifiez ni `version-stage` ni `version-id`, alors la version par défaut est la version `AWSCURRENT`.

Ce segment peut ne pas inclure le caractère deux-points (:).

version-id

L'identifiant unique de la version du secret à utiliser. Si vous spécifiez `version-id`, ne spécifiez pas `version-stage`. Si vous ne spécifiez ni `version-stage` ni `version-id`, alors la version par défaut est la version `AWSCURRENT`.

Ce segment peut ne pas inclure le caractère deux-points (:).

Exemples

Rubriques

- [Récupération des valeurs de nom d'utilisateur et de mot de passe à partir d'un secret](#)
- [Récupération de l'intégralité de SecretString](#)
- [Extraction d'une valeur d'une version spécifique d'un secret](#)
- [Récupérer les secrets d'un autre Compte AWS](#)

Récupération des valeurs de nom d'utilisateur et de mot de passe à partir d'un secret

Le [AWS::RDS::DBInstance](#) suivant récupère les valeurs du nom d'utilisateur et du mot de passe stockées dans le secret *MySecret*. Cet exemple montre le modèle recommandé pour les références

dynamiques sans version, qui utilise automatiquement la version `AWSCURRENT` et prend en charge les workflows de rotation Secrets Manager sans nécessiter de modifications du modèle.

JSON

```
{
  "MyRDSInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName": "MyRDSInstance",
      "AllocatedStorage": "20",
      "DBInstanceClass": "db.t2.micro",
      "Engine": "mysql",
      "MasterUsername":
        "{{resolve:secretsmanager:MySecret:SecretString:username}}",
      "MasterUserPassword":
        "{{resolve:secretsmanager:MySecret:SecretString:password}}"
    }
  }
}
```

YAML

```
MyRDSInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName: MyRDSInstance
    AllocatedStorage: '20'
    DBInstanceClass: db.t2.micro
    Engine: mysql
    MasterUsername: '{{resolve:secretsmanager:MySecret:SecretString:username}}'
    MasterUserPassword: '{{resolve:secretsmanager:MySecret:SecretString:password}}'
```

Récupération de l'intégralité de SecretString

La référence dynamique suivante permet de récupérer le SecretString pour *MySecret*.

```
{{resolve:secretsmanager:MySecret}}
```

Autrement :

```
{{resolve:secretsmanager:MySecret:::}}
```

Extraction d'une valeur d'une version spécifique d'un secret

La référence dynamique suivante récupère la valeur *password* de la version *AWSPREVIOUS* de *MySecret*.

```
{{resolve:secretsmanager:MySecret:SecretString:password:AWSPREVIOUS}}
```

Récupérer les secrets d'un autre Compte AWS

La référence dynamique suivante récupère le SecretString pour *MySecret* qui se trouve dans un autre Compte AWS. Vous devez spécifier l'ARN secret complet pour accéder aux secrets d'un autre Compte AWS.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret}}
```

La référence dynamique suivante récupère la valeur *password* pour *MySecret* qui se trouve dans un autre Compte AWS. Vous devez spécifier l'ARN secret complet pour accéder aux secrets d'un autre Compte AWS.

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret:SecretString:password}}
```

Obtenir des AWS valeurs à l'aide de pseudo-paramètres

Les pseudo-paramètres sont des variables intégrées qui permettent d'accéder à des informations importantes sur AWS l'environnement, telles que le compte IDs, les noms de région et les détails de la pile, qui peuvent changer entre les déploiements ou les environnements.

Vous pouvez utiliser des pseudo-paramètres au lieu de valeurs codées en dur pour rendre vos modèles plus portables et plus faciles à réutiliser dans différentes Comptes AWS régions.

Syntaxe

Vous pouvez référencer des pseudo-paramètres à l'aide de la fonction Ref intrinsèque ou de la fonction Fn::Sub intrinsèque.

Réf

La fonction intrinsèque Ref utilise la syntaxe générale suivante. Pour plus d'informations, consultez [Ref](#).

JSON

```
{ "Ref" : "AWS::PseudoParameter" }
```

YAML

```
!Ref AWS::PseudoParameter
```

Fn::And

La fonction intrinsèque Fn::Sub utilise un format différent qui inclut la syntaxe `${}` autour du pseudo-paramètre. Pour plus d'informations, consultez [Fn::Sub](#).

JSON

```
{ "Fn::Sub" : "${AWS::PseudoParameter}" }
```

YAML

```
!Sub '${AWS::PseudoParameter}
```

Pseudo-paramètres disponibles

AWS::AccountId

Renvoie l'ID du compte AWS dans lequel la pile est créée, par exemple 123456789012.

Ce pseudo-paramètre est couramment utilisé lors de la définition de rôles, de politiques et d'autres politiques de ressources IAM impliquant des informations spécifiques à un compte ARNs.

AWS::NotificationARNs

Renvoie la liste des noms de ressources Amazon (ARNs) pour les rubriques Amazon SNS qui reçoivent des notifications d'événements de pile. Vous pouvez les spécifier ARNs via l'option `notification-arns` dans AWS CLI ou via la console lorsque vous créez ou mettez à jour votre stack.

Contrairement aux autres pseudo-paramètres qui renvoient une valeur unique, `AWS::NotificationARNs` renvoie une liste de ARNs. Pour accéder à un ARN spécifique de la liste, utilisez la fonction `Fn::Select` intrinsèque. Pour de plus amples informations, veuillez consulter [Fn::Select](#).

AWS::NoValue

Supprime la propriété de ressource correspondante lorsqu'elle est spécifiée comme valeur de retour dans la fonction intrinsèque `Fn::If`. Pour de plus amples informations, veuillez consulter [Fn::If](#).

Ce pseudo-paramètre est particulièrement utile pour créer des propriétés de ressources conditionnelles qui ne doivent être incluses que dans certaines conditions.

AWS::Partition

Renvoie la partition dans laquelle se trouve la ressource. En standard Régions AWS, la partition est `aws`. Pour les ressources dans d'autres partitions, la partition est `aws-partitionname`. Par exemple, la partition pour les ressources dans les régions de Chine (Pékin et Ningxia) est `aws-cn` et la partition pour les ressources dans la région AWS GovCloud (ouest des États-Unis) est `aws-us-gov`.

La partition fait partie de l'ARN des ressources. L'utilisation `AWS::Partition` garantit le bon fonctionnement de vos modèles sur les différentes AWS partitions.

AWS::Region

Renvoie une chaîne représentant la région dans laquelle la ressource globale est créée, comme `us-west-2`.

C'est l'un des pseudo-paramètres les plus couramment utilisés, car il permet aux modèles de s'adapter aux différents Régions AWS sans modification.

AWS::StackId

Renvoie l'ID (ARN) de la pile, par exemple `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`.

AWS::StackName

Renvoie le nom de la pile, tel que `teststack`.

Le nom de la pile est couramment utilisé pour créer des noms de ressources uniques qui sont facilement identifiables comme appartenant à une pile spécifique.

AWS::URLSuffix

Renvoie le suffixe du AWS domaine dans Région AWS le quel la pile est déployée. Le suffixe est généralement `amazonaws.com`, mais pour la région Chine (Pékin), le suffixe est `amazonaws.com.cn`.

Ce paramètre est particulièrement utile lors de la construction URLs pour les points de terminaison AWS de service.

Exemples

Rubriques

- [Utilisation de base](#)
- [Utilisation de AWS::NotificationARNs](#)
- [Propriétés conditionnelles avec AWS::NoValue](#)

Utilisation de base

Les exemples suivants créent deux ressources : une rubrique Amazon SNS et une CloudWatch alarme qui envoie des notifications à cette rubrique. Ils utilisent `AWS::StackName`, `AWS::Region`, et `AWS::AccountId` pour insérer dynamiquement le nom de la pile, le courant Région AWS et l'ID de compte dans les noms des ressources, les descriptions et ARNs.

JSON

```
{
  "Resources": {
    "MyNotificationTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "DisplayName": { "Fn::Sub": "Notifications for ${AWS::StackName}" }
      }
    },
    "CPUALarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
```



```

        "AlarmDescription": { "Fn::Sub": "Alarm for high CPU in
${AWS::Region}" },
        "AlarmName": { "Fn::Sub": "${AWS::StackName}-HighCPUAlarm" },
        "MetricName": "CPUUtilization",
        "Namespace": "AWS/EC2",
        "Statistic": "Average",
        "Period": 300,
        "EvaluationPeriods": 1,
        "Threshold": 80,
        "ComparisonOperator": "GreaterThanOrEqualToThreshold",
        "AlarmActions": [{ "Fn::Sub": "arn:aws:sns:${AWS::Region}:
${AWS::AccountId}:${MyNotificationTopic}" }]
    }
}
}
}

```

YAML

```

Resources:
  MyNotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: !Sub Notifications for ${AWS::StackName}
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: !Sub Alarm for high CPU in ${AWS::Region}
      AlarmName: !Sub ${AWS::StackName}-HighCPUAlarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 300
      EvaluationPeriods: 1
      Threshold: 80
      ComparisonOperator: GreaterThanOrEqualToThreshold
      AlarmActions:
        - !Sub arn:aws:sns:${AWS::Region}:${AWS::AccountId}:${MyNotificationTopic}

```

Utilisation de AWS::NotificationARNs

Les exemples suivants configurent un groupe Auto Scaling pour envoyer des notifications en cas d'événements de lancement d'instance et d'erreurs de lancement. La configuration utilise le

`AWS::NotificationARNs` pseudo-paramètre, qui fournit une liste des rubriques Amazon SNS spécifiées lors de la ARNs création de la pile. La fonction `Fn::Select` choisit le premier ARN dans cette liste.

JSON

```
"myASG": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Properties": {
    "LaunchTemplate": {
      "LaunchTemplateId": { "Ref": "myLaunchTemplate" },
      "Version": { "Fn::GetAtt": [ "myLaunchTemplate", "LatestVersionNumber" ] }
    },
    "MaxSize": "1",
    "MinSize": "1",
    "VPCZoneIdentifier": [
      "subnetIdAz1",
      "subnetIdAz2",
      "subnetIdAz3"
    ],
    "NotificationConfigurations" : [{
      "TopicARN" : { "Fn::Select" : [ "0", { "Ref" : "AWS::NotificationARNs" } ] },
      "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH",
        "autoscaling:EC2_INSTANCE_LAUNCH_ERROR" ]
    }]
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MinSize: '1'
    MaxSize: '1'
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    NotificationConfigurations:
```

```

- TopicARN:
  Fn::Select:
    - '0'
    - Ref: AWS::NotificationARNs
  NotificationTypes:
    - autoscaling:EC2_INSTANCE_LAUNCH
    - autoscaling:EC2_INSTANCE_LAUNCH_ERROR

```

Propriétés conditionnelles avec AWS::NoValue

Les exemples suivants créent une instance de base de données Amazon RDS qui utilise un instantané uniquement si un ID d'instantané est fourni. Si la `UseDBSnapshot` condition est vraie, CloudFormation utilise la valeur du `DBSnapshotName` paramètre de la `DBSnapshotIdentifier` propriété. Si la condition est fausse, la `DBSnapshotIdentifier` propriété CloudFormation est supprimée.

JSON

```

"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
        "UseDBSnapshot",
        {"Ref" : "DBSnapshotName"},
        {"Ref" : "AWS::NoValue"}
      ]
    }
  }
}

```

YAML

MyDB:

```
Type: AWS::RDS::DBInstance
Properties:
  AllocatedStorage: '5'
  DBInstanceClass: db.t2.small
  Engine: MySQL
  EngineVersion: '5.5'
  MasterUsername:
    Ref: DBUser
  MasterUserPassword:
    Ref: DBPassword
  DBParameterGroupName:
    Ref: MyRDSParamGroup
  DBSnapshotIdentifier:
    Fn::If:
      - UseDBSnapshot
      - Ref: DBSnapshotName
      - Ref: AWS::NoValue
```

Obtenir des sorties exportées à partir d'une CloudFormation pile déployée

Lorsque vous avez plusieurs piles dans la même Compte AWS région, vous souhaitez peut-être partager des informations entre elles. Cela est utile lorsqu'une pile doit utiliser des ressources créées par une autre pile.

Par exemple, vous pouvez avoir une pile qui crée des ressources réseau, telles que des sous-réseaux et des groupes de sécurité, pour vos serveurs Web. Les autres piles qui créent les serveurs Web réels peuvent alors utiliser les ressources réseau créées par la première pile. Vous n'avez pas besoin de coder en dur la ressource IDs dans le modèle de la pile ou de la transmettre IDs en tant que paramètres d'entrée.

Pour partager des informations entre les piles, vous exportez les valeurs de sortie d'une pile et les importez dans une autre pile. Voici comment cela fonctionne :

1. Dans le modèle de la première pile (par exemple, la pile réseau), vous définissez certaines valeurs à exporter à l'aide du champ `Export` dans la section `Outputs`. Pour de plus amples informations, veuillez consulter [CloudFormation Outputssyntaxe du modèle](#).
2. Lorsque vous créez ou mettez à jour cette pile, CloudFormation exportez les valeurs de sortie pour les mettre à la disposition d'autres piles de la même pile Compte AWS et de la même région.

3. Dans le modèle de l'autre pile, vous utilisez la fonction [Fn::ImportValue](#) pour importer les valeurs exportées depuis la première pile.
4. Lorsque vous créez ou mettez à jour la deuxième pile (par exemple, la pile de serveurs Web), récupère CloudFormation automatiquement les valeurs exportées de la première pile et les utilise.

Pour obtenir une procédure pas à pas et des exemple de modèles, consultez [Référez-vous aux sorties de ressources dans une autre CloudFormation pile](#).

Exportation de valeurs de sortie de pile comparée à l'utilisation de piles imbriquées

Une pile imbriquée est une pile que vous créez dans une autre pile à l'aide de la ressource `AWS::CloudFormation::Stack`. Avec les piles imbriquées, vous déployez et gérez toutes les ressources à partir d'une même pile. Vous pouvez utiliser les sorties d'une pile du groupe de piles imbriquées comme entrées dans une autre pile du groupe. Cela est différent de l'exportation de valeurs.

Si vous voulez isoler le partage d'informations au sein d'un groupe de piles imbriquées, nous vous conseillons d'utiliser des piles imbriquées. Pour partager des informations avec d'autres piles (et pas seulement dans le groupe de piles imbriquées), il convient d'exporter les valeurs. Par exemple, vous pouvez créer une pile unique avec un sous-réseau, puis exporter son ID. Les autres piles peuvent utiliser ce sous-réseau en important son ID. Chaque pile n'a pas besoin de créer son propre sous-réseau. Tant que les piles importent l'ID du sous-réseau, vous ne pouvez pas le modifier ni le supprimer.

Pour plus d'informations sur les piles imbriquées, consultez [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).

Considérations

Les restrictions suivantes s'appliquent aux références entre piles :

- Pour chacun Compte AWS, les Export noms doivent être uniques au sein d'une région.
- Il n'est pas possible de créer des références entre piles entre différentes régions. Vous pouvez utiliser la fonction intrinsèque `Fn::ImportValue` pour importer uniquement les valeurs qui ont été exportées dans la même région.
- Pour les sorties, la valeur de la propriété Name d'un Export ne peut pas utiliser de fonctions `Ref` ou `GetAtt` qui dépendent d'une ressource.

De même, la fonction `ImportValue` ne peut pas inclure de fonctions `Ref` ou `GetAtt` qui dépendent d'une ressource.

- Lorsqu'une autre pile importe une valeur de sortie, vous ne pouvez pas supprimer la pile qui exporte la valeur de sortie ou modifier la valeur de sortie exportée. Toutes les importations doivent être supprimées avant de pouvoir supprimer la pile d'exportation ou modifier la valeur de sortie.

Affichage des valeurs de sortie exportées

Si vous avez besoin de consulter les valeurs de sortie exportées de vos piles, utilisez l'une des méthodes suivantes :

Pour afficher les valeurs de sortie exportées (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Dans le panneau de navigation gauche, sélectionnez Exportations.

Pour afficher les valeurs de sortie exportées (AWS CLI)

Utilisez la commande [list-exports](#) suivante. *us-east-1* Remplacez-le par votre Région AWS.

```
aws cloudformation list-exports --region us-east-1
```

Voici un exemple de sortie.

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-a",
      "Value": "subnet-07b410xmplddcfa03"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-b",
      "Value": "subnet-075ed3xmplebd2fb1"
    }
  ]
}
```

```
    },  
    {  
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/  
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",  
      "Name": "private-vpc-vpcid",  
      "Value": "vpc-011d7xmpl1100e9841"  
    }  
  ]  
}
```

CloudFormation affiche les noms et les valeurs des sorties exportées pour la région actuelle et la pile à partir de laquelle elles ont été exportées. Pour utiliser une valeur de sortie exportée dans le modèle d'une autre pile, vous pouvez la référencer à l'aide du nom d'exportation et de la fonction `Fn::ImportValue`.

Affichage des piles qui importent une valeur de sortie exportée

Pour supprimer ou modifier des valeurs de sortie exportées, vous devez d'abord déterminer quelles piles les importent.

Pour afficher les piles qui importent une valeur de sortie exportée, utilisez l'une des méthodes suivantes :

Pour répertorier les piles qui importent une valeur de sortie exportée (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le panneau de navigation gauche, sélectionnez Exportations.
3. Pour voir quelles piles importent une valeur d'exportation donnée, choisissez le nom d'exportation pour cette valeur d'exportation. CloudFormation affiche la page des détails de l'exportation, qui répertorie toutes les piles qui importent la valeur.

Pour répertorier les piles qui importent une valeur de sortie exportée (AWS CLI)

Utilisez la commande [list-imports](#). Remplacez *us-east-1* par votre Région AWS et *private-vpc-vpcid* par le nom de la valeur de sortie exportée.

```
aws cloudformation list-imports --region us-east-1 \  
  --export-name private-vpc-vpcid
```

CloudFormation renvoie une liste des piles qui importent la valeur.

```
{
  "Imports": [
    "my-app-stack"
  ]
}
```

Une fois que vous savez quelles piles importent une valeur exportée particulière, vous devez modifier ces piles pour supprimer les fonctions `Fn::ImportValue` qui font référence aux valeurs de sortie. Vous devez supprimer toutes les importations qui font référence aux valeurs de sortie exportées avant de pouvoir supprimer ou modifier les valeurs de sortie exportées.

Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis

Lors de la création de votre modèle, vous pouvez créer des paramètres qui obligent les utilisateurs à saisir les identifiants des AWS ressources existantes ou les paramètres de Systems Manager en utilisant les types de paramètres spécialisés fournis par CloudFormation.

Rubriques

- [Présentation de](#)
- [Exemple](#)
- [Considérations](#)
- [Types de paramètres AWS spécifiques pris en charge](#)
- [Types de paramètres Systems Manager pris en charge](#)
- [Types de paramètres Systems Manager non pris en charge](#)

Présentation de

Dans CloudFormation, vous pouvez utiliser des paramètres pour personnaliser vos piles en fournissant des valeurs d'entrée lors de la création ou de la mise à jour des piles. Cette caractéristique rend vos modèles réutilisables et flexibles dans divers scénarios.

Les paramètres sont définis dans la `Parameters` section d'un CloudFormation modèle. Chaque paramètre possède un nom et un type, et peut comporter des paramètres supplémentaires tels qu'une valeur par défaut et des valeurs autorisées. Pour de plus amples informations, veuillez consulter [CloudFormation Parameters syntaxe du modèle](#).

Le type de paramètre détermine le type de valeur que le paramètre peut accepter. Par exemple, `Number` accepte uniquement des valeurs numériques, tandis que `String` accepte des valeurs textuelles.

CloudFormation fournit plusieurs types de paramètres supplémentaires que vous pouvez utiliser dans vos modèles pour référencer les AWS ressources existantes et les paramètres de Systems Manager.

Ces types de paramètres appartiennent à deux catégories :

- **AWS-types de paramètres spécifiques** : CloudFormation fournit un ensemble de types de paramètres qui aident à détecter les valeurs non valides lors de la création ou de la mise à jour d'une pile. Lorsque vous utilisez ces types de paramètres, toute personne utilisant votre modèle doit spécifier des valeurs valides à partir Compte AWS de la région dans laquelle elle crée la pile.

S'ils utilisent le AWS Management Console, CloudFormation fournit une liste préremplie des valeurs existantes provenant de leur compte et de leur région. Ainsi, l'utilisateur n'a pas besoin de mémoriser ou de saisir correctement un nom ou un ID spécifique. Il lui suffit de sélectionner des valeurs dans une liste déroulante. Dans certains cas, il peut même rechercher des valeurs par ID, par nom ou par valeur d'une balise Name.

- **Types de paramètres de Systems Manager** : fournit CloudFormation également des types de paramètres correspondant aux paramètres existants dans le magasin de paramètres de Systems Manager. Lorsque vous utilisez ces types de paramètres, toute personne utilisant votre modèle doit spécifier une clé de magasin de paramètres comme valeur du type de paramètre Systems Manager, CloudFormation puis récupérer la dernière valeur dans le magasin de paramètres pour l'utiliser dans sa pile. Cela peut être utile lorsque vous devez fréquemment mettre à jour des applications avec de nouvelles valeurs de propriétés, telles que la nouvelle Amazon Machine Image (AMI) IDs. Pour plus d'informations sur le magasin de paramètres, consultez [Magasin de paramètres Systems Manager](#).

Une fois que vos paramètres sont définis dans la `Parameters` section, vous pouvez référencer les valeurs des paramètres dans l'ensemble de votre CloudFormation modèle à l'aide de la `Ref` fonction.

Exemple

L'exemple suivant montre un modèle utilisant les types de paramètres suivants.

- `AWS::EC2::VPC::Id`
- `AWS::EC2::Subnet::Id`

- `AWS::EC2::KeyPair::KeyName`
- `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`

Pour créer une pile à partir de ce modèle, vous devez spécifier un ID VPC, un ID de sous-réseau et un nom de paire de clés existants dans votre compte. Vous pouvez également spécifier une clé de magasin de paramètres existante qui fait référence à l'ID d'AMI souhaité ou conserver la valeur par défaut de `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`. Ce paramètre public est un alias de l'ID AMI régional le plus récent pour l'AMI Amazon Linux 2. Pour plus d'informations sur les paramètres publics, consultez [Découverte des paramètres publics dans le magasin de paramètres](#) dans le Guide de l'utilisateur AWS Systems Manager .

JSON

```
{
  "Parameters": {
    "VpcId": {
      "Description": "ID of an existing Virtual Private Cloud (VPC).",
      "Type": "AWS::EC2::VPC::Id"
    },
    "PublicSubnetId": {
      "Description": "ID of an existing public subnet within the specified VPC.",
      "Type": "AWS::EC2::Subnet::Id"
    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to enable SSH access to the instance.",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "AMIId": {
      "Description": "Name of a Parameter Store parameter that stores the ID of the Amazon Machine Image (AMI).",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    }
  },
  "Resources": {
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "VpcId": { "Ref": "VpcId" },

```

```

        "SecurityGroupIngress": [
            {
                "IpProtocol": "tcp",
                "FromPort": 22,
                "ToPort": 22,
                "CidrIp": "0.0.0.0/0"
            }
        ]
    },
    "Ec2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "KeyName": { "Ref": "KeyName" },
            "ImageId": { "Ref": "AMIID" },
            "NetworkInterfaces": [
                {
                    "AssociatePublicIpAddress": "true",
                    "DeviceIndex": "0",
                    "SubnetId": { "Ref": "PublicSubnetId" },
                    "GroupSet": [{ "Ref": "InstanceSecurityGroup" }]
                }
            ]
        }
    },
    "Outputs": {
        "InstanceId": {
            "Value": { "Ref": "Ec2Instance" }
        }
    }
}

```

YAML

Parameters:

VpcId:

Description: ID of an existing Virtual Private Cloud (VPC).

Type: AWS::EC2::VPC::Id

PublicSubnetId:

Description: ID of an existing public subnet within the specified VPC.

Type: AWS::EC2::Subnet::Id

KeyName:

```

    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance.
    Type: AWS::EC2::KeyPair::KeyName
  AMIID:
    Description: Name of a Parameter Store parameter that stores the ID of the Amazon
    Machine Image (AMI).
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
  Resources:
    InstanceSecurityGroup:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupDescription: Enable SSH access via port 22
        VpcId: !Ref VpcId
        SecurityGroupIngress:
          - IpProtocol: tcp
            FromPort: 22
            ToPort: 22
            CidrIp: 0.0.0.0/0
    Ec2Instance:
      Type: AWS::EC2::Instance
      Properties:
        KeyName: !Ref KeyName
        ImageId: !Ref AMIID
        NetworkInterfaces:
          - AssociatePublicIpAddress: "true"
            DeviceIndex: "0"
            SubnetId: !Ref PublicSubnetId
            GroupSet:
              - !Ref InstanceSecurityGroup
  Outputs:
    InstanceId:
      Value: !Ref Ec2Instance

```

AWS CLI commande pour créer la pile

La commande [create-stack](#) suivante crée une pile basée sur l'exemple de modèle.

```

aws cloudformation create-stack --stack-name MyStack \
  --template-body file:///sampletemplate.json \
  --parameters \
    ParameterKey="VpcId",ParameterValue="vpc-a123baa3" \
    ParameterKey="PublicSubnetId",ParameterValue="subnet-123a351e" \
    ParameterKey="KeyName",ParameterValue="MyKeyName" \

```

```
ParameterKey="AMIId",ParameterValue="MyParameterKey"
```

Pour utiliser un type de paramètre qui accepte une liste de chaînes, par exemple `List<AWS::EC2::Subnet::Id>`, vous devez échapper les virgules dans la `ParameterValue` à l'aide d'une double barre oblique inversée, comme illustré dans l'exemple suivant.

```
--parameters ParameterKey="SubnetIDs",ParameterValue="subnet-5ea0c127\\, subnet-6194ea3b\\, subnet-c87f2be0"
```

Considérations

Il est fortement recommandé d'utiliser des références dynamiques afin de restreindre l'accès aux définitions de configuration sensibles, comme les informations d'identification tierces. Pour de plus amples informations, veuillez consulter [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

Si vous souhaitez autoriser les utilisateurs du modèle à spécifier des valeurs différentes Comptes AWS, n'utilisez pas de types de paramètres AWS spécifiques. Définissez plutôt des paramètres de type `String` ou `CommaDelimitedList`.

Voici quelques éléments à garder à l'esprit sur les types de paramètres Systems Manager :

- Vous pouvez voir les valeurs résolues des paramètres dans l'onglet Paramètres de la pile dans la console, ou en exécutant [describe-stacks](#) ou [describe-change-set](#). Notez que ces valeurs sont définies lorsque la pile est créée ou mise à jour, elles peuvent donc différer des valeurs actuellement stockées dans le magasin de paramètres.
- Pour les mises à jour de pile, lorsque vous utilisez l'option Utiliser la valeur existante (ou si vous la définissez `UsePreviousValue` sur `true`), cela signifie que vous souhaitez continuer à utiliser la même clé de magasin de paramètres, et non sa valeur. CloudFormation récupère toujours la dernière valeur.
- Si vous spécifiez des valeurs autorisées ou d'autres contraintes, vous les CloudFormation validez par rapport aux clés de paramètres que vous spécifiez, mais pas à leurs valeurs. Vous devez valider les valeurs directement dans le magasin de paramètres.
- Lorsque vous créez ou mettez à jour des piles et que vous créez des ensembles de modifications CloudFormation, utilisez la valeur existante dans le Parameter Store à ce moment-là. Si un paramètre spécifié n'existe pas dans le Parameter Store sous celui de l'appelant Compte AWS, CloudFormation renvoie une erreur de validation.

- Lorsque vous exécutez un ensemble de modifications, CloudFormation utilise les valeurs spécifiées dans le jeu de modifications. Vous devriez consulter ces valeurs avant d'exécuter le jeu de modifications, car celles-ci peuvent changer dans Parameter Store entre la création du jeu de modifications et son exécution.
- Pour les paramètres du magasin de paramètres stockés dans celui-ci Compte AWS, vous devez fournir le nom du paramètre. Pour les paramètres du magasin de paramètres partagés par un autre Compte AWS, vous devez fournir l'ARN complet des paramètres.

Types de paramètres AWS spécifiques pris en charge

CloudFormation prend en charge les types AWS spécifiques suivants :

`AWS::EC2::AvailabilityZone::Name`

Zone de disponibilité (us-west-2a, par exemple).

`AWS::EC2::Image::Id`

ID d'image Amazon EC2 tel que `ami-0ff8a91507f77f867`. Notez que la CloudFormation console n'affiche pas de liste déroulante de valeurs pour ce type de paramètre.

`AWS::EC2::Instance::Id`

ID d'instance Amazon EC2, tel que `i-1e731a32`.

`AWS::EC2::KeyPair::KeyName`

Nom de paire de clés Amazon EC2.

`AWS::EC2::SecurityGroup::GroupName`

Nom d'un groupe de sécurité VPC par défaut, par exemple `my-sg-abc`.

`AWS::EC2::SecurityGroup::Id`

ID de groupe de sécurité (sg-a123fd85, par exemple).

`AWS::EC2::Subnet::Id`

ID de sous-réseau (subnet-123a351e, par exemple).

`AWS::EC2::Volume::Id`

ID de volume Amazon EBS, tel que `vol-3cdd3f56`.

`AWS::EC2::VPC::Id`

ID d'un VPC (vpc-a123baa3, par exemple).

`AWS::Route53::HostedZone::Id`

ID de zone hébergée Amazon Route 53, tel que Z23YXV40VPL04A.

`List<AWS::EC2::AvailabilityZone::Name>`

Tableau de zones de disponibilité d'une région (us-west-2a, us-west-2b, par exemple).

`List<AWS::EC2::Image::Id>`

Un tableau d'images Amazon EC2 IDs, tel que. ami-0ff8a91507f77f867, ami-0a584ac55a7631c0c Notez que la CloudFormation console n'affiche pas de liste déroulante de valeurs pour ce type de paramètre.

`List<AWS::EC2::Instance::Id>`

Un tableau d'instances Amazon EC2 IDs, tel que. i-1e731a32, i-1e731a34

`List<AWS::EC2::SecurityGroup::GroupName>`

Tableau de noms de groupes de sécurité VPC par défaut, comme my-sg-abc, my-sg-def.

`List<AWS::EC2::SecurityGroup::Id>`

Un ensemble de groupes de sécurité IDs, tels que sg-a123fd85, sg-b456fd85.

`List<AWS::EC2::Subnet::Id>`

Un tableau de sous-réseaux IDs, tel que subnet-123a351e, subnet-456b351e.

`List<AWS::EC2::Volume::Id>`

Un tableau de volumes Amazon EBS IDs, tel que vol-3cdd3f56, vol-4cdd3f56.

`List<AWS::EC2::VPC::Id>`

Un ensemble de VPC IDs, tel que. vpc-a123baa3, vpc-b456baa3

`List<AWS::Route53::HostedZone::Id>`

Un ensemble de zones hébergées Amazon Route 53 IDs, telles que Z23YXV40VPL04A, Z23YXV40VPL04B.

Types de paramètres Systems Manager pris en charge

CloudFormation prend en charge les types de paramètres Systems Manager suivants :

`AWS::SSM::Parameter::Name`

Nom d'une clé de paramètre Systems Manager. Utilisez ce type de paramètre uniquement pour vérifier l'existence d'un paramètre obligatoire. CloudFormation ne récupérera pas la valeur réelle associée au paramètre.

`AWS::SSM::Parameter::Value<String>`

Un paramètre Systems Manager dont la valeur est une chaîne. C'est le type de paramètre `String` dans Parameter Store.

`AWS::SSM::Parameter::Value<List<String>>` ou

`AWS::SSM::Parameter::Value<CommaDelimitedList>`

Un paramètre Systems Manager dont la valeur est une liste de chaînes. C'est le type de paramètre `StringList` dans Parameter Store.

`AWS::SSM::Parameter::Value<AWS-specific parameter type>`

Paramètre Systems Manager dont la valeur est un type de paramètre AWS spécifique.

L'exemple suivant spécifie le type `AWS::EC2::KeyPair::KeyName` :

- `AWS::SSM::Parameter::Value<AWS::EC2::KeyPair::KeyName>`

`AWS::SSM::Parameter::Value<List<AWS-specific parameter type>>`

Paramètre Systems Manager dont la valeur est une liste de types de paramètres AWS spécifiques.

L'exemple suivant spécifie une liste de types `AWS::EC2::KeyPair::KeyName` :

- `AWS::SSM::Parameter::Value<List<AWS::EC2::KeyPair::KeyName>>`

Types de paramètres Systems Manager non pris en charge

CloudFormation ne prend pas en charge le type de paramètre Systems Manager suivant :

- Listes des types de paramètres Systems Manager, par exemple :

`List<AWS::SSM::Parameter::Value<String>>`

En outre, CloudFormation ne prend pas en charge la définition de paramètres de modèle en tant que types de paramètres SecureString Systems Manager. Cependant, vous pouvez spécifier des chaînes sécurisées comme valeurs de paramètres pour certaines ressources. Pour de plus amples informations, veuillez consulter [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

CloudFormation procédures pas à pas

Cette documentation fournit une collection de démonstrations pas à pas conçues pour vous permettre de mettre en pratique le déploiement de piles.

- [Référez-vous aux sorties de ressources dans une autre CloudFormation pile](#)— Cette procédure pas à pas vous montre comment référencer les sorties d'une CloudFormation pile au sein d'une autre pile. Au lieu d'inclure toutes les ressources dans une seule pile, vous pouvez créer des AWS ressources connexes dans des piles distinctes afin de créer des modèles plus modulaires et réutilisables.
- [Déploiement d'applications sur Amazon EC2](#)— Découvrez comment l'utiliser CloudFormation pour installer, configurer et démarrer automatiquement votre application sur les EC2 instances Amazon. De cette manière, vous pouvez facilement dupliquer les déploiements et mettre à jour les installations existantes sans vous connecter directement aux instances.
- [Mettre à jour une CloudFormation pile](#)— Suivez la progression simple des mises à jour d'une pile en cours d'exécution avec CloudFormation.
- [Création d'une application redimensionnée et équilibrée de charge](#)— Découvrez comment CloudFormation créer une application évolutive et équilibrée en termes de charge. Ce guide pratique couvre la création d'un groupe Auto Scaling, d'un équilibreur de charge et d'autres ressources associées afin de garantir que votre application puisse gérer des charges de trafic variables et maintenir une haute disponibilité.
- [Homologue avec un VPC dans un autre Compte AWS](#)— Cette procédure pas à pas vous guide tout au long du processus de création d'une connexion d'appairage Virtual Private Cloud (VPC) entre deux personnes de manière différente. VPCs Comptes AWS Le peering VPC vous aide à acheminer le trafic entre les ressources VPCs et les ressources d'accès comme si elles faisaient partie du même réseau.
- [Effectuez des blue/green déploiements ECS en utilisant CodeDeploy CloudFormation](#)— Découvrez comment les utiliser pour CloudFormation effectuer des AWS CodeDeploy blue/green deployments on Amazon ECS. Blue/green déploiements sont un moyen de mettre à jour vos applications ou services avec un minimum de temps d'arrêt.

Référez-vous aux sorties de ressources dans une autre CloudFormation pile

Cette procédure pas à pas vous montre comment référencer les sorties d'une CloudFormation pile au sein d'une autre pile afin de créer des modèles plus modulaires et réutilisables.

Au lieu d'inclure toutes les ressources dans une seule pile, vous créez des AWS ressources connexes dans des piles distinctes. Vous pouvez ensuite référencer les sorties de ressources requises à partir d'autres piles. En limitant les références entre piles aux sorties, vous contrôlez les parties d'une pile qui sont référencées par d'autres piles.

Par exemple, supposons que vous disposez d'une pile de réseau constituée d'un VPC, d'un groupe de sécurité, d'un sous-réseau pour les applications web publiques et d'une pile d'applications web publiques distincte. Pour garantir que les applications web publiques utilisent le groupe de sécurité et le sous-réseau de la pile de réseau, vous devez créer une référence entre piles qui permette à la pile d'applications web de faire référence à des sorties de ressource de la pile de réseau. Avec une référence entre piles, les propriétaires des piles d'applications web n'ont pas besoin de créer ou de gérer des règles de mise en réseau ou des ressources.

Pour créer une référence entre piles, utilisez le champ de sortie `Export` pour signaler la valeur d'une sortie de ressource pour l'exportation. Ensuite, utilisez la fonction intrinsèque `Fn::ImportValue` pour importer la valeur. Pour de plus amples informations, veuillez consulter [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#).

Note

CloudFormation est un service gratuit. Toutefois, les AWS ressources que vous incluez dans vos piles vous sont facturées au taux en vigueur pour chacune d'entre elles. Pour plus d'informations sur la tarification AWS, consultez [la page de détails de chaque produit](#).

Rubriques

- [Utilisez un modèle d'exemple pour créer une pile réseau](#)
- [Utilisation d'un modèle d'exemple pour créer une pile d'applications Web](#)
- [Vérification du bon fonctionnement de la pile](#)
- [Dépannage des erreurs de mappage AMI](#)
- [Nettoyage de vos ressources](#)

Utilisez un modèle d'exemple pour créer une pile réseau

Avant de commencer cette procédure pas à pas, vérifiez que vous disposez des autorisations IAM pour utiliser tous les services suivants : Amazon VPC, Amazon EC2 et CloudFormation

La pile de réseau contient le VPC, le groupe de sécurité et le sous-réseau que vous souhaitez utiliser dans la pile d'applications web. En plus de ces ressources, la pile de réseau crée une passerelle Internet et des tables de routage pour autoriser l'accès public.

Vous devez créer cette pile avant de créer la pile d'applications web. Si vous créez la pile d'applications web en premier, elle ne contient pas de groupe de sécurité ni de sous-réseau.

Le modèle de pile est disponible à l'adresse URL suivante : <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template>. Pour voir les ressources créées par la pile, cliquez sur le lien, qui ouvre le modèle. Dans la section Outputs, vous pouvez voir les ressources réseau que le modèle d'exemple exporte. Les noms des ressources exportées sont préfixés du nom de la pile dans le cas où vous exportez des ressources de mise en réseau à partir d'autres piles. Lorsque les utilisateurs importent des ressources de mise en réseau, ils peuvent indiquer à partir de quelle pile les ressources sont importées.

Pour créer la pile de réseau

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Piles, choisissez Créer une pile, puis Avec de nouvelles ressources (standard).
3. Sélectionnez Choisir un modèle existant, puis dans la section Spécifier le modèle, sélectionnez URL Amazon S3.
4. Pour URL Amazon S3, collez l'URL suivante : **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**.
5. Choisissez Suivant.
6. Pour Nom de la pile, tapez **SampleNetworkCrossStack**, puis choisissez Suivant.

Note

Notez le nom de cette pile. Vous en aurez besoin lorsque vous lancerez la pile d'applications web.

7. Choisissez Suivant. Pour cette procédure, vous n'avez pas besoin d'ajouter de balises ni de spécifier des paramètres avancés.
8. Assurez-vous que le nom de la pile et l'URL du modèle sont corrects, puis choisissez Créer une pile.

La création de votre pile peut prendre plusieurs minutes. CloudFormation Attendez que toutes les ressources aient été créées avant de procéder à la création de la pile d'applications web.

9. Pour surveiller la progression, affichez les événements de la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Utilisation d'un modèle d'exemple pour créer une pile d'applications Web

La pile d'applications web crée une instance EC2 qui utilise le groupe de sécurité et le sous-réseau de la pile de réseau.

Vous devez créer cette pile de la même manière Région AWS que la pile réseau.

Le modèle de pile est disponible à l'adresse URL suivante : <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>. Pour voir les ressources que la pile créera, cliquez sur ce lien, qui ouvre le modèle. Dans la section Ressources, affichez les propriétés de l'instance EC2. Vous pouvez voir comment les ressources de mise en réseau sont importées à partir d'une autre pile à l'aide de la fonction Fn::ImportValue.

Pour créer la pile d'applications web

1. Depuis la page CloudFormation, choisissez Piles, puis Créer une pile avec de nouvelles ressources (standard).
2. Sélectionnez Choisir un modèle existant, puis dans la section Spécifier le modèle, sélectionnez URL Amazon S3.
3. Pour URL Amazon S3, collez l'URL suivante : **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**.
4. Choisissez Suivant.
5. Pour le Nom de la pile, tapez **SampleWebAppCrossStack**. Dans la section Paramètres, utilisez la valeur par défaut du NetworkStackNameparamètre, puis choisissez Next.

L'exemple de modèle utilise la valeur du paramètre pour indiquer à partir de quelle pile les valeurs sont importées.

6. Choisissez Suivant. Pour cette procédure, vous n'avez pas besoin d'ajouter de balises ni de spécifier des paramètres avancés.
7. Assurez-vous que le nom de la pile et l'URL du modèle sont corrects, puis choisissez Créer une pile.

La création de votre pile peut prendre plusieurs minutes. CloudFormation

Vérification du bon fonctionnement de la pile

Une fois que la pile a été créée, affichez ses ressources et notez l'ID d'instance. Pour plus d'informations sur l'affichage des ressources d'une pile, consultez la page [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

Pour vérifier le groupe de sécurité et le sous-réseau de l'instance, examinez les propriétés de l'instance dans la [console Amazon EC2](#). Si l'instance utilise le groupe de sécurité et le sous-réseau de la pile SampleNetworkCrossStack, cela signifie que la référence entre piles a bien été créée.

Servez-vous de la console pour examiner les sorties de pile et l'URL de l'exemple de site web afin de vérifier que l'application web est bien en cours d'exécution. Pour de plus amples informations, veuillez consulter [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

Dépannage des erreurs de mappage AMI

Si vous recevez l'erreur `Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64`, cela signifie que le modèle ne comprend pas de mappage AMI pour votre Région AWS. Au lieu de mettre à jour le mappage, nous vous recommandons d'utiliser les paramètres publics de Systems Manager pour référencer dynamiquement les dernières versions AMIs :

1. Téléchargez le SampleWebAppCrossStack modèle sur votre machine locale depuis : <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>.
2. Supprimez l'intégralité de la section de mappage AWSRegionArch2AMI.
3. Ajoutez le paramètre Systems Manager suivant :

```
"LatestAmiId": {
  "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
  "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
  "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
}
```

4. Remplacez la référence ImageId existante :

```
"ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" } ,
  "HVM64" ] },
```

par :

```
"ImageId": { "Ref": "LatestAmiId" },
```

Ce paramètre résout automatiquement la dernière AMI Amazon Linux 2 pour la région dans laquelle vous déployez la pile.

Pour les autres distributions Linux, utilisez le chemin d'accès au paramètre approprié. Pour plus d'informations, consultez [Découverte des paramètres publics dans le stocker de paramètres](#) du Guide de l'utilisateur AWS Systems Manager .

5. Chargez le modèle modifié dans un compartiment S3 de votre compte :

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. Lors de la création de la pile, indiquez l'URL de votre modèle S3 à la place de l'URL d'exemple.

Nettoyage de vos ressources

Pour éviter que des services non souhaités vous soient facturés, supprimez les piles.

Pour supprimer les piles

1. Dans la CloudFormation console, choisissez la SampleWebAppCrossStack pile.
2. Choisissez Actions, puis Supprimer la pile.
3. Dans le message de confirmation, choisissez Supprimer.

4. Une fois que la pile a été supprimée, répétez les mêmes étapes pour la pile `SampleNetworkCrossStack`.

Note

Attendez que la `SampleWebAppCrossStack` pile soit CloudFormation complètement supprimée. Si l'instance EC2 est toujours en cours d'exécution dans le VPC CloudFormation, le VPC ne sera pas supprimé de la pile `SampleNetworkCrossStack`.

Déploiement d'applications sur Amazon EC2

Vous pouvez l'utiliser CloudFormation pour installer, configurer et démarrer automatiquement des applications sur des instances Amazon EC2. Cette approche vous permet de dupliquer facilement les déploiements et de mettre à jour les installations existantes sans avoir à vous connecter directement à l'instance. Vous évitez ainsi de perdre du temps et de l'énergie.

CloudFormation inclut un ensemble de scripts d'assistance (`cfn-init`, `cfn-signal`, `cfn-get-metadata`, etc `cfn-hup`) basés sur `cloud-init`. Vous appelez ces scripts d'assistance depuis vos CloudFormation modèles pour installer, configurer et mettre à jour des applications sur des instances Amazon EC2 qui se trouvent dans le même modèle. Pour plus d'informations, consultez la [référence aux scripts CloudFormation d'assistance](#) dans le Guide de référence des CloudFormation modèles.

Dans le [tutoriel de démarrage](#), vous avez créé un simple serveur web en utilisant UserData avec un script bash de base. Bien que cela fonctionne pour une simple page « Hello World », les applications réelles exigent souvent une configuration plus sophistiquée, incluant :

- Installation de plusieurs packages logiciels dans le bon ordre.
- Création de fichiers de configuration complexes avec un contenu précis.
- Démarrage de services et configuration pour une exécution automatique.
- Gestion des erreurs et validation du processus d'installation.

CloudFormation les scripts d'assistance fournissent un moyen plus robuste et plus facile à maintenir de configurer les instances EC2 par rapport aux scripts bash de base dans. UserData Le script d'assistance `cfn-init` lit les données de configuration dans les métadonnées de votre modèle et les applique de manière systématique à votre instance.

Dans ce tutoriel, vous apprendrez à utiliser le script d'assistance `cfn-init` et à surveiller le processus d'amorçage.

Note

CloudFormation est gratuit, mais les ressources Amazon EC2 que vous créez vous seront facturées. Toutefois, si vous êtes nouveau dans ce AWS domaine, vous pouvez profiter du [niveau gratuit](#) pour minimiser ou éliminer les coûts au cours de ce processus d'apprentissage.

Rubriques

- [Conditions préalables](#)
- [Concepts fondamentaux de l'amorçage](#)
- [Exemple d'amorçage simple](#)
- [Ajout de fichiers et de commandes](#)
- [Ajout de la sécurité réseau](#)
- [Modèle complet d'amorçage](#)
- [Création de la pile à l'aide de la console](#)
- [Surveillance du processus d'amorçage](#)
- [Test du serveur web amorcé](#)
- [Résolution des problèmes d'amorçage](#)
- [nettoyer des ressources ;](#)
- [Étapes suivantes](#)

Conditions préalables

- Vous devez avoir suivi le [Création de votre première pile](#) didacticiel ou avoir une expérience équivalente des CloudFormation notions de base.
- Vous devez avoir accès à un utilisateur ou Compte AWS à un rôle IAM autorisé à utiliser Amazon EC2 CloudFormation et/ou à accéder à un utilisateur administratif.
- Vous devez disposer d'un cloud privé virtuel (VPC) ayant accès à Internet. Ce modèle de tutoriel nécessite un VPC par défaut, qui est automatiquement fourni avec les environnements Comptes AWS plus récents. Si vous ne disposez pas d'un VPC par défaut ou s'il a été supprimé,

consultez la section de résolution des problèmes dans le tutoriel [Création de votre première pile](#) pour trouver d'autres solutions.

Concepts fondamentaux de l'amorçage

Avant de créer le modèle, examinons les concepts clés qui permettent le fonctionnement de l'amorçage.

le script d'assistant **cfn-init**.

CloudFormation fournit des scripts d'assistance Python que vous pouvez utiliser pour installer des logiciels et démarrer des services sur une instance Amazon EC2. Le script `cfn-init` lit les métadonnées de ressource dans votre modèle et applique la configuration à votre instance.

Le processus fonctionne comme suit :

1. Vous définissez la configuration dans la section Metadata de votre ressource EC2.
2. Vous appelez `cfn-init` depuis le script UserData.
3. `cfn-init` lit les métadonnées et applique la configuration.
4. Votre instance est configurée selon vos spécifications.

Structure des métadonnées

La configuration est définie dans une structure spécifique au sein de votre instance EC2.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:                                # Metadata section for the resource
      AWS::CloudFormation::Init:            # Required key that cfn-init looks for
        config:                             # Configuration name (you can have multiple)
          packages:                          # Install packages
          files:                             # Create files
          commands:                          # Run commands
          services:                          # Start/stop services
```

Le script `cfn-init` traite ces sections dans un ordre précis : packages, groupes, utilisateurs, sources, fichiers, commandes, puis services.

Exemple d'amorçage simple

Commençons par un exemple minimal d'amorçage qui installe et démarre Apache.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:                # Install Apache web server
            yum:
              httpd: []
          services:                # Start Apache and enable it to start on boot
            systemd:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64           # Script that runs when instance starts
      Fn::Sub: |
        #!/bin/bash
        yum install -y aws-cfn-bootstrap
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
        region ${AWS::Region}
```

Cet exemple simple illustre les concepts de base :

- La section `packages` installe le package `httpd` à l'aide de `yum`. Cela fonctionne sur Amazon Linux et sur d'autres distributions Linux utilisant `yum`.
- La section `services` s'assure que `httpd` démarre et s'exécute automatiquement.
- `UserData` installe les outils d'amorçage les plus récents et appelle `cfn-init`.

Ajout de fichiers et de commandes

Poursuivons l'exemple en ajoutant une page web personnalisée et un fichier journal dans le répertoire `/var/log` de l'instance EC2.

Création de fichiers

La section `files` vous permet de créer des fichiers sur l'instance avec un contenu précis. Le caractère vertical (`|`) permet de transmettre un bloc de texte littéral (le code HTML) comme contenu du fichier (`/var/www/html/index.html`).

```
files:
  /var/www/html/index.html:
    content: |
      <body>
        <h1>Congratulations, you have successfully launched the AWS CloudFormation
sample.</h1>
      </body>
```

Exécution des commandes

La section `commands` permet d'exécuter des commandes shell pendant le processus d'amorçage. Cette commande crée un fichier journal à `/var/log/welcome.txt` sur l'instance EC2. Pour le consulter, vous devez disposer d'une paire de clés Amazon EC2 pour l'accès SSH, ainsi qu'une plage d'adresses IP autorisée à se connecter en SSH à l'instance (non couvert ici).

```
commands:
  createWelcomeLog:
    command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

Ajout de la sécurité réseau

Comme nous configurons un serveur web, nous devons autoriser le trafic web (HTTP) à atteindre notre instance EC2. Pour ce faire, nous allons créer un groupe de sécurité qui autorise le trafic entrant sur le port 80 depuis votre adresse IP. Les instances EC2 doivent également pouvoir envoyer du trafic vers l'internet, par exemple pour installer des mises à jour de packages. Par défaut, les groupes de sécurité autorisent tout le trafic sortant. Nous associerons ensuite ce groupe de sécurité à notre instance EC2 à l'aide de la propriété `SecurityGroupIds`.

```
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow HTTP access from my IP address
    SecurityGroupIngress:
      - IpProtocol: tcp
        Description: HTTP
```

```

FromPort: 80
ToPort: 80
CidrIp: !Ref MyIP

```

Modèle complet d'amorçage

Maintenant, assemblons toutes les pièces. Voici le modèle complet qui combine tous les concepts que nous avons abordés.

```

AWSTemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.?\d{1,2}$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP access from my IP address
      SecurityGroupIngress:
        - IpProtocol: tcp

```

```
    Description: HTTP
    FromPort: 80
    ToPort: 80
    CidrIp: !Ref MyIP

WebServer:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
        files:
          /var/www/html/index.html:
            content: |
              <body>
                <h1>Congratulations, you have successfully launched the AWS
CloudFormation sample.</h1>
              </body>
        commands:
          createWelcomeLog:
            command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
      services:
        systemd:
          httpd:
            enabled: true
            ensureRunning: true
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    UserData: !Base64
      Fn::Sub: |
        #!/bin/bash
        yum install -y aws-cfn-bootstrap
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
    Tags:
      - Key: Name
        Value: Bootstrap Tutorial Web Server

Outputs:
```

```
WebsiteURL:
  Value: !Sub 'http://${WebServer.PublicDnsName}'
  Description: EC2 instance public DNS name
```

Création de la pile à l'aide de la console

La procédure suivante implique le téléchargement du modèle de pile d'échantillons à partir d'un fichier. Ouvrez un éditeur de texte sur votre ordinateur local et ajoutez le modèle. Enregistrez le fichier sous le nom `samplelinux2stack.template`.

Pour lancer la pile (modèle)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sélectionnez Créer une pile, Avec de nouvelles ressources (standard).
3. Dans Spécifier le modèle, choisissez Charger un fichier de modèle, puis Choisir un fichier pour charger le fichier `samplelinux2stack.template`.
4. Choisissez Suivant.
5. Sur la page Spécifier les détails de la pile, saisissez **BootstrapTutorialStack** comme nom de pile.
6. Pour Paramètres, procédez comme suit.
 - LatestAmild: conservez la valeur par défaut.
 - InstanceType: Choisissez t2.micro ou t3.micro pour le type d'instance EC2.
 - MyIP : saisissez votre adresse IP publique avec un suffixe /32.
7. Choisissez Suivant deux fois, puis Soumettre pour créer la pile.

Surveillance du processus d'amorçage

Les processus d'amorçage prennent plus de temps que les lancements EC2 simples, car des logiciels supplémentaires doivent être installés et configurés.

Surveillance de la progression de l'amorçage

1. Dans la CloudFormation console, sélectionnez votre stack et ouvrez l'onglet Events.
2. Surveillez l'événement `WebServer CREATE_IN_PROGRESS`. Le processus d'amorçage commence après le lancement de l'instance.

3. Le processus d'amorçage dure généralement quelques minutes. Vous verrez `WebServer CREATE_COMPLETE` lorsqu'il sera terminé.

Si vous souhaitez voir ce qui se passe pendant le processus d'amorçage, vous pouvez consulter les journaux de l'instance.

Consultation des journaux d'amorçage (facultatif)

1. Ouvrez la [console EC2](#) et trouvez votre instance.
2. Sélectionnez l'instance, puis choisissez Actions, Surveiller et résoudre les problèmes, Obtenir le journal du système afin d'afficher la progression de l'amorçage.
3. Si les journaux ne s'affichent pas immédiatement, attendez un moment, puis actualisez la page.

Test du serveur web amorcé

Lorsque votre pile indique `CREATE_COMPLETE`, testez votre serveur web.

Pour tester le serveur Web

1. Dans la CloudFormation console, accédez à l'onglet Sorties de votre stack.
2. Cliquez sur la valeur `WebsiteURL` pour ouvrir votre serveur web dans un nouvel onglet.
3. Vous devriez voir votre page web personnalisée affichant le message `Congratulations, you have successfully launched the AWS CloudFormation sample.`

Note

Si la page ne se charge pas immédiatement, attendez une minute et réessayez. Le processus d'amorçage peut encore être en cours même si la pile indique `CREATE_COMPLETE`.

Résolution des problèmes d'amorçage

Si votre processus d'amorçage échoue ou si votre serveur web ne fonctionne pas, voici les problèmes courants et leurs solutions.

Problèmes courants

- Échec de la création de la pile : consultez l'onglet Événements pour voir les messages d'erreur précis.
- Serveur web inaccessible : vérifiez que votre adresse IP est correcte dans le paramètre MyIP. N'oubliez pas d'ajouter /32 à la fin.
- Échec du processus d'amorçage : l'instance peut être lancée, mais `cfn-init` échoue. Consultez les journaux du système comme décrit dans la section sur la surveillance.

nettoyer des ressources ;

Pour éviter les frais continus, vous pouvez effectuer un nettoyage en supprimant la pile et ses ressources.

Pour supprimer la pile et ses ressources

1. Ouvrez la [CloudFormation console](#).
2. Sur la page Piles, sélectionnez l'option à côté du nom de la pile que vous avez créée (**BootstrapTutorialStack**), puis choisissez Supprimer.
3. Lorsque vous êtes invité à confirmer l'opération, choisissez Supprimer.
4. Surveillez la progression du processus de suppression de la pile dans l'onglet Événement. L'état de **BootstrapTutorialStack** est remplacé par DELETE_IN_PROGRESS. Lorsque la suppression de la pile est CloudFormation terminée, elle supprime la pile de la liste.

Étapes suivantes

Félicitations ! Vous avez appris avec succès comment démarrer des instances EC2 avec CloudFormation Vous comprenez maintenant :

- Comment utiliser les scripts d'assistance `cfn-init`
- Comment structurer les métadonnées pour l'amorçage
- Comment installer des packages, créer des fichiers, exécuter des commandes et gérer des services
- Comment surveiller les problèmes d'amorçage

Pour continuer à apprendre :

- Apprenez à mettre à jour une pile en cours d'exécution et à utiliser le script d'`cfn-hup` assistance. Pour de plus amples informations, veuillez consulter [Mettre à jour une CloudFormation pile](#).
- Apprenez à amorcer une pile Windows. Pour de plus amples informations, veuillez consulter [Stacks basés sur le bootstrap Windows CloudFormation](#).
- Explorez des scénarios d'amorçage plus complexes avec plusieurs ensembles de configuration. Pour plus d'informations, consultez [cfn-init](#) et le guide de [AWS::CloudFormation::Init](#) référence des CloudFormation modèles.
- Découvrez comment utiliser `cfn-signal` pour le rapport d'état d'achèvement de l'amorçage. Pour plus d'informations, consultez [cfn-signal](#) dans le Guide de référence des modèles CloudFormation .

Mettre à jour une CloudFormation pile

Note

Ce didacticiel s'appuie sur les concepts du [Déploiement d'applications sur Amazon EC2](#) didacticiel. Si vous n'avez pas terminé ce didacticiel, nous vous recommandons de le faire d'abord pour comprendre le EC2 bootstrap avec CloudFormation

Cette rubrique décrit la progression simple des mises à jour d'une pile en cours d'exécution. Nous effectuerons les étapes suivantes :

1. Création de la pile initiale : créez une pile à l'aide d'une AMI Amazon Linux 2 de base, en installant l'application PHP Apache Web Server et une simple application PHP à l'aide des scripts d'CloudFormation assistance.
2. Mettre à jour l'application : mettez à jour l'un des fichiers de l'application et déployez le logiciel à l'aide de CloudFormation.
3. Ajouter une paire de clés : ajoutez une paire de EC2 clés Amazon à l'instance, puis mettez à jour le groupe de sécurité pour autoriser l'accès SSH à l'instance.
4. Mettre à jour le type d'instance : modifiez le type d'instance de l' EC2 instance Amazon sous-jacente.
5. Mettre à jour l'AMI : modifiez l'image machine Amazon (AMI) de l' EC2 instance Amazon de votre stack.

 Note

CloudFormation est gratuit, mais les EC2 ressources Amazon que vous créez vous seront facturées. Toutefois, si vous êtes nouveau dans ce AWS domaine, vous pouvez profiter du [niveau gratuit](#) pour minimiser ou éliminer les coûts au cours de ce processus d'apprentissage.

Rubriques

- [Étape 1 : Création de la pile initiale](#)
- [Étape 2 : mise à jour de l'application](#)
- [Étape 3 : ajouter un accès SSH avec une paire de clés](#)
- [Étape 4 : mettre à jour le type d'instance](#)
- [Étape 5 : mise à jour de l'AMI](#)
- [Considérations en matière d'impact et de disponibilité](#)
- [Ressources connexes](#)

Étape 1 : Création de la pile initiale

Nous allons commencer par créer une pile que nous pourrions utiliser dans le reste de cette rubrique. Nous avons fourni un modèle simple qui lance une application Web PHP à instance unique hébergée Apache Web Server et exécutée sur une AMI Amazon Linux 2.

PHP et l'application PHP simple sont tous installés par les scripts d' CloudFormation assistance installés par défaut sur l'AMI Amazon Linux 2. Apache Web Server L'extrait de modèle suivant montre les métadonnées qui décrivent les packages et les fichiers à installer, en l'occurrence l'infrastructure PHP Apache Web Server et l'infrastructure PHP du Yum référentiel pour l'AMI Amazon Linux 2. L'extrait montre également la Services section, ce qui garantit que le Apache Web Server est en cours d'exécution.

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
```

```

    php: []
  files:
    /var/www/html/index.php:
      content: |
        <?php
        echo '<h1>Hello World!</h1>';
        ?>
      mode: '000644'
      owner: apache
      group: apache
  services:
    systemd:
      httpd:
        enabled: true
        ensureRunning: true

```

L'application elle-même est un exemple de « Hello World » entièrement défini dans le modèle. Pour une application réelle, les fichiers peuvent être stockés sur Amazon S3 ou dans un autre référentiel et référencés à partir du modèle. GitHub CloudFormation peut télécharger des packages (tels que RPMs ou RubyGems) et référencer des fichiers individuels, des extensions .zip et .tar des fichiers pour créer les artefacts de l'application sur l' EC2instance Amazon.

Le modèle active et configure le `cfn-hup` daemon pour qu'il écoute les modifications apportées à la configuration définie dans les métadonnées de l'instance Amazon EC2 . À l'aide du `cfn-hup` daemon, vous pouvez mettre à jour le logiciel d'application, tel que la version d'Apache ou de PHP, ou vous pouvez mettre à jour le fichier d'application PHP lui-même à partir de CloudFormation. L'extrait suivant, tiré de la même EC2 ressource Amazon dans le modèle, montre les éléments nécessaires à la configuration pour appeler `cfn-init` toutes les deux minutes `cfn-hup` afin de détecter et d'appliquer les mises à jour aux métadonnées. Sinon, il `cfn-init` ne s'exécute qu'une seule fois au démarrage.

```

files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack=${AWS::StackId}
      region=${AWS::Region}
      # The interval used to check for changes to the resource metadata in minutes.
      Default is 15
      interval=2
      mode: '000400'

```

```
owner: root
group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
content: !Sub |
    [cfn-auto-reloader-hook]
    triggers=post.update
    path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
    action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --region
    ${AWS::Region}
    runas=root
services:
  systemd:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
```

Pour compléter la pile, dans la Properties section de définition de l' EC2 instance Amazon, la UserData propriété contient le cloud-init script qui appelle cfn-init pour installer les packages et les fichiers. Pour plus d'informations, consultez la [référence aux scripts CloudFormation d'assistance](#) dans le Guide de référence des CloudFormation modèles. Le modèle crée également un groupe EC2 de sécurité Amazon.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

```
    Type: String
```

```
    Default: t3.micro
```

```
    AllowedValues:
```

- t3.nano
- t3.micro
- t3.small
- t3.medium
- t3a.nano

- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

yum update -y aws-cfn-bootstrap

Run cfn-init

/opt/aws/bin/cfn-init -v --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region} || error_exit 'Failed to run cfn-init'

Start up the cfn-hup daemon to listen for changes to the EC2 instance

metadata

/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'

Signal success or failure

/opt/aws/bin/cfn-signal -e \$? --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region}

Metadata:

```

AWS::CloudFormation::Init:
  config:
    packages:
      yum:
        httpd: []
        php: []
    files:
      /var/www/html/index.php:
        content: |
          <?php
            echo "<h1>Hello World!</h1>";
          ?>
        mode: '000644'
        owner: apache
        group: apache
      /etc/cfn/cfn-hup.conf:
        content: !Sub |
          [main]
          stack=${AWS::StackId}
          region=${AWS::Region}
          # The interval used to check for changes to the resource metadata in
minutes. Default is 15
          interval=2
          mode: '000400'
          owner: root
          group: root
      /etc/cfn/hooks.d/cfn-auto-reloader.conf:
        content: !Sub |
          [cfn-auto-reloader-hook]
          triggers=post.update
          path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
          action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
          runas=root
    services:
      systemd:
        httpd:
          enabled: true
          ensureRunning: true
        cfn-hup:
          enabled: true
          ensureRunning: true
          files:
            - /etc/cfn/cfn-hup.conf

```

```
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
CreationPolicy:
  ResourceSignal:
    Timeout: PT5M

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
    Description: URL of the web application
```

Pour lancer une pile à partir de ce modèle

1. Copiez le modèle et enregistrez-le localement sur votre système sous forme de fichier texte. Notez son emplacement, car vous en aurez besoin pour utiliser le fichier dans une étape ultérieure.
2. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
3. Choisissez Créer une pile, avec de nouvelles ressources (standard).
4. Choisissez Choisir un modèle existant.
5. Sous Spécifier le modèle, choisissez Télécharger un fichier modèle et naviguez jusqu'au fichier que vous avez créé lors de la première étape, puis choisissez Suivant.
6. Sur la page Spécifier les détails de la pile, entrez le **UpdateTutorial** nom de la pile.
7. Sous Paramètres, conservez tous les mêmes paramètres et choisissez Next deux fois.
8. Sur l'écran Réviser et créer, choisissez Soumettre.

Une fois que le statut de votre stack est CREATE_COMPLETE atteint, l'onglet Sorties affiche l'URL de votre site Web. Si vous choisissez la valeur de la WebsiteURL sortie, vous verrez votre nouvelle application PHP fonctionner.

Étape 2 : mise à jour de l'application

Maintenant que nous avons déployé la pile, nous allons mettre à jour l'application. Nous allons apporter une modification basique du texte qui est affiché par l'application. Pour ce faire, nous allons ajouter une commande echo dans le fichier index.php, comme illustré dans cet extrait de modèle :

```
files:
  /var/www/html/index.php:
    content: |
      <?php
      echo "<h1>Hello World!</h1>";
      echo "<p>This is an updated version of our application.</p>";
      ?>
    mode: '000644'
    owner: apache
    group: apache
```

Utilisez un éditeur de texte pour modifier manuellement le fichier de modèle que vous avez enregistré localement.

Maintenant, mettez à jour la pile.

Pour mettre à jour la pile avec votre modèle mis à jour

1. Dans la CloudFormation console, sélectionnez votre **UpdateTutorial** stack.
2. Choisissez Mettre à jour, puis effectuer une mise à jour directe.
3. Choisissez Remplacer le modèle existant.
4. Sous Spécifier le modèle, choisissez Télécharger un fichier modèle et téléchargez votre fichier modèle modifié, puis cliquez sur Suivant.
5. Sur la page Spécifier les détails de la pile, conservez tous les paramètres identiques et cliquez deux fois sur Suivant.
6. Sur la page Révision, passez en revue les modifications. Sous Modifications, vous devriez voir que la WebServerInstance ressource CloudFormation sera mise à jour.
7. Sélectionnez Soumettre.

Lorsque votre pile est dans UPDATE_COMPLETE cet état, vous pouvez choisir à nouveau la valeur de WebsiteURL sortie pour vérifier que les modifications apportées à votre application ont pris effet. Le

cfn-hup daemon s'exécute toutes les 2 minutes. L'application peut donc prendre jusqu'à 2 minutes pour changer une fois la pile mise à jour.

Pour voir l'ensemble des ressources mises à jour, accédez à la CloudFormation console. Dans l'onglet Événements, examinez les événements de la pile. Dans ce cas particulier, les métadonnées de l' EC2 instance Amazon ont `WebServerInstance` été mises à jour, ce qui a également entraîné CloudFormation la réévaluation des autres ressources (`WebServerSecurityGroup`) pour s'assurer qu'il n'y avait pas d'autres modifications. Aucune des autres ressources de pile n'a été modifiée. CloudFormation met uniquement à jour les ressources qui sont concernées par les modifications de la pile. Ces modifications peuvent être directes, telles que des modifications de propriétés ou de métadonnées, ou elles peuvent être dues à des dépendances ou à des flux de données `RefGetAtt`, ou à d'autres fonctions intrinsèques du modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

Cette simple mise à jour illustre le processus. Cependant, vous pouvez apporter des modifications beaucoup plus complexes aux fichiers et aux packages déployés sur vos EC2 instances Amazon. Par exemple, vous pouvez choisir d'ajouter MySQL à l'instance, ainsi que la prise en charge PHP pour MySQL. Pour ce faire, il suffit d'ajouter les packages, les fichiers et les services supplémentaires à la configuration, puis de mettre à jour la pile pour déployer les modifications.

```
packages:
  yum:
    httpd: []
    php: []
    mysql: []
    php-mysql: []
    mysql-server: []
    mysql-libs: []

    ...

services:
  systemd:
    httpd:
      enabled: true
      ensureRunning: true
    cfn-hup:
      enabled: true
      ensureRunning: true
    files:
      - /etc/cfn/cfn-hup.conf
```

```
- /etc/cfn/hooks.d/cfn-auto-reloader.conf
mysqld:
  enabled: true
  ensureRunning: true
```

Vous pouvez mettre à jour les CloudFormation métadonnées pour les mettre à jour avec les nouvelles versions des packages utilisés par l'application. Dans les exemples précédents, la propriété de version de chaque package est vide, ce qui indique que la dernière version du package `cfn-init` doit être installée.

```
packages:
  yum:
    httpd: []
    php: []
```

Vous pouvez éventuellement spécifier une chaîne de version d'un package. Si vous modifiez cette chaîne de version dans les appels ultérieurs de mise à jour de la pile, la nouvelle version du package sera déployée. Voici un exemple d'utilisation des numéros de version pour les RubyGems packages. Tous les packages compatibles avec la gestion des versions peuvent inclure des versions spécifiques.

```
packages:
  rubygems:
    mysql: []
    rubygems-update:
      - "1.6.2"
    rake:
      - "0.8.7"
    rails:
      - "2.3.11"
```

Étape 3 : ajouter un accès SSH avec une paire de clés

Vous pouvez également mettre à jour une ressource dans le modèle pour ajouter des propriétés qui n'étaient pas initialement spécifiées dans le modèle. Pour illustrer cela, nous allons ajouter une paire de EC2 clés Amazon à une EC2 instance existante, puis ouvrir le port 22 dans le groupe de EC2 sécurité Amazon afin que vous puissiez utiliser Secure Shell (SSH) pour accéder à l'instance.

Pour ajouter un accès SSH à une instance Amazon EC2 existante

1. Ajoutez deux paramètres supplémentaires au modèle pour transmettre le nom d'une paire de EC2 clés Amazon et d'un emplacement SSH existants.

Parameters:

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

2. Ajoutez la KeyName propriété à l' EC2 instance Amazon.

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref WebServerSecurityGroup

3. Ajoutez le port 22 et l'emplacement SSH aux règles d'entrée pour le groupe de EC2 sécurité Amazon.

WebServerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Enable HTTP access via port 80 and SSH access via port 22

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

```
ToPort: 80
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
  FromPort: 22
  ToPort: 22
  CidrIp: !Ref SSHLocation
```

4. Mettez à jour la pile en suivant les mêmes étapes que celles expliquées dans [Étape 2 : mise à jour de l'application](#).

Étape 4 : mettre à jour le type d'instance

Démontrons maintenant comment mettre à jour l'infrastructure sous-jacente en modifiant le type d'instance.

La pile que nous avons construite jusqu'à présent utilise une instance Amazon EC2 t3.micro. Supposons que votre site Web nouvellement créé reçoit plus de trafic qu'une instance t3.micro ne peut en gérer, et que vous souhaitiez maintenant passer à un type d'instance Amazon EC2 m5.large. Si l'architecture du type d'instance change, l'instance doit être créée avec une autre AMI. Cependant, le t3.micro et le m5.large utilisent les mêmes architectures de processeur et exécutent Amazon Linux 2 (x86_64). AMIs Pour plus d'informations, consultez la section [Compatibilité pour la modification du type d'instance](#) dans le guide de EC2 l'utilisateur Amazon.

Utilisons le modèle que nous avons modifié à l'étape précédente pour changer le type d'instance. Comme InstanceType il s'agissait d'un paramètre d'entrée du modèle, nous n'avons pas besoin de modifier le modèle ; nous pouvons modifier la valeur du paramètre sur la page Spécifier les détails de la pile.

Pour mettre à jour la pile avec une nouvelle valeur de paramètre

1. Dans la CloudFormation console, sélectionnez votre **UpdateTutorial** stack.
2. Choisissez Mettre à jour, puis effectuer une mise à jour directe.
3. Choisissez Utiliser un modèle existant, puis cliquez sur Suivant.
4. Sur la page Spécifier les détails de la pile, remplacez la valeur de la zone t3.micro de InstanceType par m5.large. Choisissez ensuite deux fois Next.
5. Sur la page Révision, passez en revue les modifications. Sous Modifications, vous devriez voir que la WebServerInstance ressource CloudFormation sera mise à jour.
6. Sélectionnez Soumettre.

Vous pouvez modifier dynamiquement le type d'instance d'une instance Amazon basée sur EBS en démarrant et en arrêtant l' EC2 instance. CloudFormation essaie d'optimiser le changement en mettant à jour le type d'instance et en redémarrant l'instance, afin que l'ID de l'instance ne change pas. Toutefois, au redémarrage de l'instance, l'adresse IP publique de cette dernière change. Pour garantir que l'adresse IP élastique est correctement liée après la modification, elle CloudFormation sera également mise à jour. Vous pouvez voir les modifications dans la CloudFormation console sous l'onglet Événements.

Pour vérifier le type d'instance depuis le AWS Management Console, ouvrez la EC2 console Amazon et localisez-y votre instance.

Étape 5 : mise à jour de l'AMI

Mettons maintenant à jour notre stack pour utiliser Amazon Linux 2023, qui est la prochaine génération d'Amazon Linux.

La mise à jour de l'AMI est une modification majeure qui nécessite le remplacement de l'instance. Nous ne pouvons pas simplement démarrer et arrêter l'instance pour modifier l'AMI ; CloudFormation cela représente une modification d'une propriété immuable de la ressource. Pour modifier une propriété immuable, vous CloudFormation devez lancer une ressource de remplacement, en l'occurrence une nouvelle EC2 instance Amazon exécutant la nouvelle AMI.

Voyons comment nous pourrions mettre à jour notre modèle de stack pour utiliser Amazon Linux 2023. Les principales modifications incluent la mise à jour du paramètre AMI et le passage du gestionnaire de `dnf` packages `yum` au gestionnaire de packages.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2023 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

```
    Type: String
```

```
    Default: t3.micro
```

```
    AllowedValues:
```

```
      - t3.nano
```

```
      - t3.micro
```

- t3.small
- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\/\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

```

KeyName: !Ref KeyName
SecurityGroupIds:
  - !Ref WebServerSecurityGroup
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    # Get the latest CloudFormation package
    dnf update -y aws-cfn-bootstrap
    # Run cfn-init
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region} || error_exit 'Failed to run cfn-init'

    # Start up the cfn-hup daemon to listen for changes to the EC2 instance
metadata
  /opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
  # Signal success or failure
  /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        dnf:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            echo "<p>This is an updated version of our application.</p>";
            echo "<p>Running on Amazon Linux 2023!</p>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2

```

```
    mode: '000400'
    owner: root
    group: root
  /etc/cfn/hooks.d/cfn-auto-reloader.conf:
    content: !Sub |
      [cfn-auto-reloader-hook]
      triggers=post.update
      path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
      action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
      runas=root
  services:
    systemd:
      httpd:
        enabled: true
        ensureRunning: true
      cfn-hup:
        enabled: true
        ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  CreationPolicy:
    ResourceSignal:
      Timeout: PT5M

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
```


Description: URL of the web application

Mettez à jour la pile en suivant les mêmes étapes que celles expliquées dans [Étape 2 : mise à jour de l'application](#).

Une fois la nouvelle instance en cours d'exécution, CloudFormation met à jour les autres ressources de la pile pour qu'elles pointent vers la nouvelle ressource. Lorsque toutes les nouvelles ressources sont créées, l'ancienne ressource est supprimée, un processus appelé UPDATE_CLEANUP. Cette fois-ci, vous remarquerez que l'ID d'instance et l'URL d'application de l'instance dans la pile ont changé suite à la mise à jour. Les événements du tableau des événements contiennent une description « La mise à jour demandée entraîne la modification d'une propriété immuable et crée donc une nouvelle ressource physique » pour indiquer qu'une ressource a été remplacée.

Alternativement : si le code d'application que vous souhaitez mettre à jour est écrit dans l'AMI, vous pouvez utiliser le même mécanisme de mise à jour de pile pour mettre à jour l'AMI afin de charger votre nouvelle application.

Pour mettre à jour l'AMI avec un code d'application personnalisé

1. Créez votre nouvelle AMI contenant les modifications apportées à votre application ou à votre système d'exploitation. Pour plus d'informations, consultez la section [Créer une AMI basée sur Amazon EBS dans le guide](#) de l'utilisateur Amazon. EC2
2. Mettez à jour votre modèle pour intégrer le nouvel ID d'AMI.
3. Mettez à jour la pile en suivant les mêmes étapes que celles expliquées dans [Étape 2 : mise à jour de l'application](#).

Lorsque vous mettez à jour la pile, vous CloudFormation détectez que l'ID de l'AMI a changé, puis cela déclenche une mise à jour de la pile de la même manière que nous avons initié la mise à jour ci-dessus.

Considérations en matière d'impact et de disponibilité

Différentes propriétés ont des impacts distincts sur les ressources de la pile. Vous pouvez utiliser CloudFormation pour mettre à jour n'importe quelle propriété. Toutefois, avant d'effectuer des modifications, posez-vous les questions suivantes :

1. Comment la mise à jour affectera-t-elle la ressource elle-même ? Par exemple, la mise à jour d'un seuil d'alarme désactivera l'alarme pendant la mise à jour. Comme nous l'avons vu, la modification du type d'instance nécessite l'arrêt et le redémarrage de l'instance. CloudFormation

utilise les actions de mise à jour ou de modification des ressources sous-jacentes pour apporter des modifications aux ressources. Pour comprendre l'impact des mises à jour, consultez la documentation des ressources spécifiques.

2. La modification est-elle réversible ou immuable ? Certaines modifications apportées aux propriétés des ressources, telles que la modification de l'AMI sur une EC2 instance Amazon, ne sont pas prises en charge par les services sous-jacents. Dans le cas de modifications modifiables, le type Update ou Modify CloudFormation sera utilisé APIs pour les ressources sous-jacentes. Pour les modifications de propriétés immuables, CloudFormation créera de nouvelles ressources avec les propriétés mises à jour, puis les liera à la pile avant de supprimer les anciennes ressources. Bien que l'on CloudFormation essaie de réduire le temps d'arrêt des ressources de la pile, le remplacement d'une ressource est un processus en plusieurs étapes qui prend du temps. Au cours de la reconfiguration de la pile, votre application ne sera pas entièrement opérationnelle. Par exemple, il ne peut-être qu'elle ne parvienne pas à traiter les requêtes ou à accéder à une base de données.

Ressources connexes

Pour plus d'informations sur l'utilisation CloudFormation pour démarrer des applications et sur l'intégration à d'autres services de configuration et de déploiement tels que Puppet etOpscode Chef, consultez les livres blancs suivants :

- [Démarrage des applications via CloudFormation](#)
- [Intégration CloudFormation avec Opscode Chef](#)
- [Intégration CloudFormation avec Puppet](#)

Création d'une application redimensionnée et équilibrée de charge

Pour cette procédure, vous devez créer une pile qui vous aide à configurer une application redimensionnée et équilibrée de charge. Un exemple de code et un exemple de modèle sont mis à votre disposition pour créer la pile. L'exemple de modèle fournit un groupe Auto Scaling, une Application Load Balancer, des groupes de sécurité qui contrôlent le trafic vers l'équilibreur de charge et vers le groupe Auto Scaling, ainsi qu'une configuration de notification Amazon SNS pour publier des notifications concernant les activités de dimensionnement.

Ce modèle crée au moins une instance Amazon EC2 et une Application Load Balancer. Vous serez facturé pour les ressources AWS utilisées si vous créez une pile à partir de ce modèle.

Modèle de pile complète

Commençons par le modèle.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  InstanceType:
    Description: The EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues:
      - t3.micro
      - t3.small
      - t3.medium
  KeyName:
    Description: Name of an existing EC2 key pair to allow SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
  OperatorEmail:
    Description: The email address to notify when there are any scaling activities
    Type: String
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  Subnets:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: At least two public subnets in different Availability Zones in the
selected VPC
  VPC:
    Type: AWS::EC2::VPC::Id
    Description: A virtual private cloud (VPC) that enables resources in public subnets
to connect to the internet
Resources:
  ELBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
```

Properties:

GroupDescription: ELB Security Group

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

EC2SecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: EC2 Security Group

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

SourceSecurityGroupId:

Fn::GetAtt:

- ELBSecurityGroup

- GroupId

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: !Ref SSHLocation

EC2TargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 30

HealthCheckProtocol: HTTP

HealthCheckTimeoutSeconds: 15

HealthyThresholdCount: 5

Matcher:

HttpCode: '200'

Name: EC2TargetGroup

Port: 80

Protocol: HTTP

TargetGroupAttributes:

- Key: deregistration_delay.timeout_seconds

Value: '20'

UnhealthyThresholdCount: 3

VpcId: !Ref VPC

ALBListener:

Type: AWS::ElasticLoadBalancingV2::Listener

```
Properties:
  DefaultActions:
    - Type: forward
      TargetGroupArn: !Ref EC2TargetGroup
  LoadBalancerArn: !Ref ApplicationLoadBalancer
  Port: 80
  Protocol: HTTP
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    Subnets: !Ref Subnets
    SecurityGroups:
      - !GetAtt ELBSecurityGroup.GroupId
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      SecurityGroupIds:
        - !Ref EC2SecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd
          systemctl enable httpd
          echo "<h1>Hello World!</h1>" > /var/www/html/index.html
NotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
WebServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref LaunchTemplate
      Version: !GetAtt LaunchTemplate.LatestVersionNumber
```

```

    MaxSize: '3'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref NotificationTopic
        NotificationTypes: ['autoscaling:EC2_INSTANCE_LAUNCH',
'autoscaling:EC2_INSTANCE_LAUNCH_ERROR', 'autoscaling:EC2_INSTANCE_TERMINATE',
'autoscaling:EC2_INSTANCE_TERMINATE_ERROR']
    TargetGroupARNs:
      - !Ref EC2TargetGroup
    VPCZoneIdentifier: !Ref Subnets

```

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "InstanceType": {
      "Description": "The EC2 instance type",
      "Type": "String",
      "Default": "t3.micro",
      "AllowedValues": [
        "t3.micro",
        "t3.small",
        "t3.medium"
      ]
    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to allow SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "LatestAmiId": {
      "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "OperatorEmail": {
      "Description": "The email address to notify when there are any scaling activities",
      "Type": "String"
    },
    "SSHLocation": {

```

```
    "Description": "The IP address range that can be used to SSH to the EC2
instances",
    "Type": "String",
    "MinLength": 9,
    "MaxLength": 18,
    "Default": "0.0.0.0/0",
    "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "Subnets": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "At least two public subnets in different Availability Zones in the
selected VPC"
  },
  "VPC": {
    "Type": "AWS::EC2::VPC::Id",
    "Description": "A virtual private cloud (VPC) that enables resources in public
subnets to connect to the internet"
  }
},
"Resources": {
  "ELBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "ELB Security Group",
      "VpcId": {
        "Ref": "VPC"
      }
    },
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
      }
    ]
  },
  "EC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "EC2 Security Group",
      "VpcId": {
        "Ref": "VPC"
      }
    }
  }
}
```

```
"SecurityGroupIngress":[
  {
    "IpProtocol":"tcp",
    "FromPort":80,
    "ToPort":80,
    "SourceSecurityGroupId":{
      "Fn::GetAtt":[
        "ELBSecurityGroup",
        "GroupId"
      ]
    }
  },
  {
    "IpProtocol":"tcp",
    "FromPort":22,
    "ToPort":22,
    "CidrIp":{
      "Ref":"SSHLocation"
    }
  }
],
},
"EC2TargetGroup":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":30,
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":15,
    "HealthyThresholdCount":5,
    "Matcher":{
      "HttpCode":"200"
    },
    "Name":"EC2TargetGroup",
    "Port":80,
    "Protocol":"HTTP",
    "TargetGroupAttributes":[
      {
        "Key":"deregistration_delay.timeout_seconds",
        "Value":"20"
      }
    ],
    "UnhealthyThresholdCount":3,
    "VpcId":{
```



```

        "Ref": "VPC"
    }
}
},
"ALBListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
        "DefaultActions": [
            {
                "Type": "forward",
                "TargetGroupArn": {
                    "Ref": "EC2TargetGroup"
                }
            }
        ],
        "LoadBalancerArn": {
            "Ref": "ApplicationLoadBalancer"
        },
        "Port": 80,
        "Protocol": "HTTP"
    }
},
"ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
        "Scheme": "internet-facing",
        "Subnets": {
            "Ref": "Subnets"
        },
        "SecurityGroups": [
            {
                "Fn::GetAtt": [
                    "ELBSecurityGroup",
                    "GroupId"
                ]
            }
        ]
    }
},
"LaunchTemplate": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
        "LaunchTemplateName": {
            "Fn::Sub": "${AWS::StackName}-launch-template"
        }
    }
}

```

```

    },
    "LaunchTemplateData":{
      "ImageId":{
        "Ref":"LatestAmiId"
      },
      "InstanceType":{
        "Ref":"InstanceType"
      },
      "KeyName":{
        "Ref":"KeyName"
      },
      "SecurityGroupIds":[
        {
          "Ref":"EC2SecurityGroup"
        }
      ],
      "UserData":{
        "Fn::Base64":{
          "Fn::Join":[
            "",
            [
              "#!/bin/bash\n",
              "yum update -y\n",
              "yum install -y httpd\n",
              "systemctl start httpd\n",
              "systemctl enable httpd\n",
              "echo \"<h1>Hello World!</h1>\n" > /var/www/html/index.html"
            ]
          ]
        }
      }
    }
  },
  "NotificationTopic":{
    "Type":"AWS::SNS::Topic",
    "Properties":{
      "Subscription":[
        {
          "Endpoint":{
            "Ref":"OperatorEmail"
          },
          "Protocol":"email"
        }
      ]
    }
  }
}

```

```

    ]
  },
  "WebServerGroup":{
    "Type":"AWS::AutoScaling::AutoScalingGroup",
    "Properties":{
      "LaunchTemplate":{
        "LaunchTemplateId":{
          "Ref":"LaunchTemplate"
        },
        "Version":{
          "Fn::GetAtt":[
            "LaunchTemplate",
            "LatestVersionNumber"
          ]
        }
      },
      "MaxSize":"3",
      "MinSize":"1",
      "NotificationConfigurations":[
        {
          "TopicARN":{
            "Ref":"NotificationTopic"
          },
          "NotificationTypes":[
            "autoscaling:EC2_INSTANCE_LAUNCH",
            "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
            "autoscaling:EC2_INSTANCE_TERMINATE",
            "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
          ]
        }
      ],
      "TargetGroupARNs":[
        {
          "Ref":"EC2TargetGroup"
        }
      ],
      "VPCZoneIdentifier":{
        "Ref":"Subnets"
      }
    }
  }
}

```

```
}
```

Description du modèle

La première partie de ce modèle spécifie le `Parameters`. Une valeur doit être affectée à chaque paramètre au moment de l'exécution CloudFormation pour approvisionner correctement la pile. Les ressources spécifiées ultérieurement dans le modèle font référence à ces valeurs et utilisent les données.

- `InstanceType` : le type d'instance EC2 provisionnée par Amazon EC2 Auto Scaling. À défaut de précision, la valeur par défaut `t3.micro` est utilisée.
- `KeyName` : une paire de clés EC2 existante pour permettre l'accès SSH aux instances.
- `LatestAmiId` : l'Amazon Machine Image (AMI) de l'instance. Si ce n'est pas spécifié, vos instances sont lancées avec une AMI Amazon Linux 2, à l'aide d'un paramètre AWS Systems Manager public géré par AWS. Pour plus d'informations, veuillez consulter la rubrique [Résultat des paramètres publics](#) dans le Guide de l'utilisateur AWS Systems Manager .
- `OperatorEmail` : l'adresse e-mail à laquelle vous souhaitez envoyer des notifications d'activité dimensionnée.
- `SSHLocation` : la plage d'adresses IP pouvant être utilisée pour se connecter aux instances en SSH.
- `Subnets` : au moins deux sous-réseaux dans des zones de disponibilité différentes.
- `VPC` : un cloud privé virtuel (VPC) de votre compte qui permet aux ressources des sous-réseaux publics de se connecter à Internet.

Note

Vous pouvez utiliser le VPC et les sous-réseaux par défaut pour autoriser les instances à accéder à Internet. Si vous utilisez votre propre VPC, assurez-vous qu'il dispose d'un sous-réseau mappé à chaque zone de disponibilité de la région dans laquelle vous travaillez. Au minimum, vous devez disposer de deux sous-réseaux publics disponibles pour créer l'équilibreur de charge.

La partie suivante de ce modèle spécifie le `Resources`. Cette section spécifie les ressources de la pile et leurs propriétés.

[AWS::EC2::SecurityGroup](#)ressource ELBSecurityGroup

- `SecurityGroupIngress` contient une règle d'entrée TCP qui autorise l'accès depuis toutes les adresses IP (« `CidrIp` ": « `0.0.0.0/0`") sur le port 80.

[AWS::EC2::SecurityGroup](#) ressource `EC2SecurityGroup`

- `SecurityGroupIngress` contient deux règles d'entrée : 1) une règle d'entrée TCP qui permet l'accès SSH (port 22) à partir de la plage d'adresses IP que vous spécifiez pour le paramètre `SSHLocation` d'entrée et 2) une règle d'entrée TCP qui permet d'accéder à l'équilibreur de charge en spécifiant le groupe de sécurité de l'équilibreur de charge. La [GetAtt](#) fonction est utilisée pour obtenir l'ID du groupe de sécurité avec le nom logique `ELBSecurityGroup`.

[AWS::ElasticLoadBalancingV2::TargetGroup](#) ressource `EC2TargetGroup`

- `Port`, `Protocol` et `HealthCheckProtocol` spécifiez le port de l'instance EC2 (80) et le protocole (HTTP) vers lesquels `ApplicationLoadBalancer` acheminer le trafic et qu'Elastic Load Balancing utilise pour vérifier l'état des instances EC2.
- `HealthCheckIntervalSeconds` spécifie que les instances EC2 ont un intervalle de 30 secondes entre la surveillance de l'état. La `HealthCheckTimeoutSeconds` désigne la durée qu'Elastic Load Balancing peut attendre pour recevoir une réponse de la cible de surveillance de l'état (15 secondes, dans cet exemple). Au delà de ce délai, Elastic Load Balancing marque la surveillance de l'état de l'instance EC2 comme étant défectueuse. Lorsque la vérification de l'état d'une instance EC2 échoue au moins cinq fois consécutives (`UnhealthyThresholdCount`), Elastic Load Balancing cesse d'y acheminer le trafic tant que la vérification de l'état d'une instance EC2 (`HealthyThresholdCount`). À ce stade, Elastic Load Balancing considère que l'instance est saine et recommence à acheminer le trafic vers l'instance.
- `TargetGroupAttributes` met à jour la valeur du délai de désinscription du groupe cible à 20 secondes. Par défaut, Elastic Load Balancing attend 300 secondes avant de terminer le processus de désinscription (vidange de la connexion).

[AWS::ElasticLoadBalancingV2::Listener](#) ressource `ALBListener`

- `DefaultActions` spécifie le port que l'équilibreur de charge écoute, le groupe cible sur lequel l'équilibreur de charge transfère les demandes et le protocole utilisé pour router les demandes.

[AWS::ElasticLoadBalancingV2::LoadBalancer](#) ressource `ApplicationLoadBalancer`

- Subnets prend la valeur du paramètre Subnets d'entrée comme la liste des sous-réseaux publics dans lesquels les nœuds d'équilibrage de charge seront créés.
- SecurityGroup obtient l'ID du groupe de sécurité qui agit en tant que pare-feu virtuel pour vos nœuds d'équilibreur de charge afin de contrôler le trafic entrant. La [GetAtt](#) fonction est utilisée pour obtenir l'ID du groupe de sécurité avec le nom logique ELBSecurityGroup.

[AWS::EC2::LaunchTemplate](#) ressource LaunchTemplate

- ImageId prend la valeur du paramètre LatestAmiId d'entrée comme AMI à utiliser.
- KeyName utilise la valeur du paramètre d'entrée KeyName comme paire de clés EC2 à utiliser.
- SecurityGroupIds obtient l'ID du groupe de sécurité avec le nom logique EC2SecurityGroup qui agit en tant que pare-feu virtuel pour vos instances EC2 afin de contrôler le trafic entrant.
- UserData est un script de configuration qui s'exécute une fois que l'instance est opérationnelle. Dans cet exemple, le script installe Apache et crée un fichier index.html.

[AWS::SNS::Topic](#) ressource NotificationTopic

- Subscription prend la valeur du paramètre OperatorEmail d'entrée comme adresse e-mail du destinataire des notifications en cas d'activités de dimensionnement.

AWS::AutoScaling::AutoScalingResource de [groupe](#) WebServerGroup

- MinSize et MaxSize définissent le nombre minimum et maximum d'instances EC2 dans le groupe Auto Scaling.
- TargetGroupARNs prend l'ARN du groupe cible avec le nom logique EC2TargetGroup. Au fur et à mesure que ce groupe Auto Scaling évolue, il enregistre et annule automatiquement les instances auprès de ce groupe cible.
- VPCZoneIdentifier prend la valeur du paramètre d'entrée Subnets comme la liste des sous-réseaux publics dans lesquels les instances EC2 peuvent être créées.

Étape 1 : Lancement la pile

Avant de lancer la pile, vérifiez que vous disposez des autorisations Gestion des identités et des accès AWS (IAM) nécessaires pour utiliser tous les services suivants : Amazon EC2, Amazon EC2 AWS Systems Manager Auto Scaling, Elastic Load Balancing, Amazon SNS et. CloudFormation

La procédure suivante implique le téléchargement du modèle de pile d'échantillons à partir d'un fichier. Ouvrez un éditeur de texte sur votre ordinateur local et ajoutez l'un des modèles. Enregistrez le fichier sous le nom `sampleloadbalancedappstack.template`.

Pour lancer la pile (modèle)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sélectionnez Créer une pile, Avec de nouvelles ressources (standard).
3. Dans Spécifier le modèle, choisissez Charger un fichier de modèle, puis Choisir un fichier pour charger le fichier `sampleloadbalancedappstack.template`.
4. Choisissez Suivant.
5. Dans la page Spécifier les détails, pour Nom de la pile, tapez un nom pour la pile (par exemple **SampleLoadBalancedAppStack**).
6. Sous Paramètres, passez en revue les paramètres de la pile et fournissez des valeurs pour tous les paramètres qui n'ont pas de valeurs par défaut, y compris OperatorEmailSSHLocation,, KeyName, VPC et sous-réseaux.
7. Choisissez Next (Suivant) deux fois.
8. Sur la page Vérification, vérifiez et confirmez les paramètres.
9. Sélectionnez Soumettre.

Vous pouvez consulter l'état de la pile dans la CloudFormation console dans la colonne État. Une CloudFormation fois la pile créée avec succès, vous recevez le statut `CREATE_COMPLETE`.

 Note

Après avoir créé la pile, vous devez confirmer l'abonnement avant que l'adresse e-mail puisse recevoir des notifications. Pour plus d'informations, veuillez consulter la rubrique [Obtenir des notifications Amazon SNS lors de la mise à l'échelle d'un groupe Auto Scaling](#) dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Étape 2 : nettoyage des exemples de ressources

Pour vous assurer de ne pas être facturé pour des ressources d'échantillons inutilisés, supprimez la pile.

Pour supprimer la pile

1. Dans la CloudFormation console, sélectionnez la SampleLoadBalancedAppStackpile.
2. Sélectionnez Delete (Supprimer).
3. Dans le message de confirmation, choisissez Supprimer la pile.

État des SampleLoadBalancedAppStackmodifications apportées à DELETE_IN_PROGRESS.
Lorsque la suppression de la pile est CloudFormation terminée, elle supprime la pile de la liste.

Utilisez les exemples de modèles de cette procédure pour créer vos propres modèles de piles. Pour de plus amples informations, consultez [Tutoriel : Configurer une application redimensionnée et à charge équilibrée](#) dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Homologue avec un VPC dans un autre Compte AWS

Vous pouvez établir un pair avec un Virtual Private Cloud (VPC) dans un autre en utilisant. Compte AWS [AWS::EC2::VPCPeeringConnection](#) Cela crée une connexion réseau entre deux personnes VPCs qui vous permet d'acheminer le trafic entre elles afin qu'elles puissent communiquer comme si elles faisaient partie du même réseau. Une connexion d'appairage de VPC peut faciliter l'accès aux données et le transfert des données.

Pour établir une connexion d'appairage VPC, vous devez autoriser deux connexions distinctes au Comptes AWS sein d'une même pile. CloudFormation

Pour plus d'informations sur l'appairage VPC et ses limitations, consultez le [Guide d'appairage Amazon VPC](#).

Conditions préalables

1. Vous avez besoin d'un identifiant VPC homologue, d'un Compte AWS identifiant homologue et d'un [rôle d'accès entre comptes](#) pour la connexion d'appairage.

 Note

Cette procédure fait référence à deux comptes : un compte qui permet l'appairage entre comptes (le compte de l'accepteur). Et un compte qui demande la connexion d'appairage (le compte du demandeur).

2. Pour accepter la connexion d'appairage VPC, vous devez assumer le rôle pour l'accès entre comptes. La ressource se comporte de la même manière qu'une ressource de connexion d'appairage de VPC dans le même compte. Pour plus d'informations sur la manière dont un administrateur IAM accorde des autorisations pour assumer le rôle inter-comptes, consultez la section [Accorder à un utilisateur des autorisations pour changer de rôle](#) dans le Guide de l'utilisateur IAM.

Étape 1 : Créer un VPC et un rôle entre comptes

Dans cette étape, vous allez créer le VPC et le rôle dans le compte de l'accepteur.

Pour créer un VPC et un rôle d'accès intercompte

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Depuis la page CloudFormation, choisissez Piles, puis Créer une pile avec de nouvelles ressources (standard).
3. Dans Prérequis – Préparer le modèle, choisissez Choisir un modèle existant, puis Charger un fichier modèle, puis Choisir un fichier.
4. Ouvrez un éditeur de texte sur votre ordinateur local et ajoutez l'un des modèles suivants. Enregistrez le fichier et revenez à la console pour le sélectionner comme fichier modèle.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  }
}
```

```
},
"Resources": {
  "vpc": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
      "CidrBlock": "10.1.0.0/16",
      "EnableDnsSupport": false,
      "EnableDnsHostnames": false,
      "InstanceTenancy": "default"
    }
  },
  "peerRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [
          {
            "Principal": {
              "AWS": {
                "Ref": "PeerRequesterAccountId"
              }
            },
            "Action": [
              "sts:AssumeRole"
            ],
            "Effect": "Allow"
          }
        ]
      },
      "Path": "/",
      "Policies": [
        {
          "PolicyName": "root",
          "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
              {
                "Effect": "Allow",
                "Action": "ec2:AcceptVpcPeeringConnection",
                "Resource": "*"
              }
            ]
          }
        }
      ]
    }
  }
}
```

```

    ]
  }
}
},
"Outputs": {
  "VPCId": {
    "Value": {
      "Ref": "vpc"
    }
  },
  "RoleARN": {
    "Value": {
      "Fn::GetAtt": [
        "peerRole",
        "Arn"
      ]
    }
  }
}
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Principal:
              AWS: !Ref PeerRequesterAccountId

```

```
        Action:
          - 'sts:AssumeRole'
        Effect: Allow
    Path: /
    Policies:
      - PolicyName: root
        PolicyDocument:
          Version: 2012-10-17
          Statement:
            - Effect: Allow
              Action: 'ec2:AcceptVpcPeeringConnection'
              Resource: '*'

Outputs:
  VPCId:
    Value: !Ref vpc
  RoleARN:
    Value: !GetAtt
      - peerRole
      - Arn
```

5. Choisissez Suivant.
6. Donnez un nom à la pile (par exemple, **VPC-owner**), puis entrez l'ID du compte AWS du compte du demandeur dans le `PeerRequesterAccountId` champ.
7. Acceptez les valeurs par défaut, puis choisissez Suivant.
8. Choisissez Je reconnais que cela CloudFormation pourrait créer des ressources IAM, puis choisissez Create stack.

Étape 2 : créer un modèle qui inclut **AWS::EC2::VPCPeeringConnection**

Maintenant que vous avez créé le VPC et le rôle entre comptes, vous pouvez établir un pair avec le VPC en utilisant un autre Compte AWS (le compte du demandeur).

Pour créer un modèle qui inclut la ressource [AWS::EC2::VPCPeeringConnection](#)

1. Retournez à la page d'accueil de la CloudFormation console.
2. Depuis la page CloudFormation, choisissez Piles, puis Créer une pile avec de nouvelles ressources (standard).
3. Dans Prérequis – Préparer le modèle, choisissez Choisir un modèle existant, puis Charger un fichier modèle, puis Choisir un fichier.

4. Ouvrez un éditeur de texte sur votre ordinateur local et ajoutez l'un des modèles suivants. Enregistrez le fichier et revenez à la console pour le sélectionner comme fichier modèle.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and a VPC Peering connection using the PeerRole to accept.",
  "Parameters": {
    "PeerVPCAccountId": {
      "Type": "String"
    },
    "PeerVPCId": {
      "Type": "String"
    },
    "PeerRoleArn": {
      "Type": "String"
    }
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.2.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "vpcPeeringConnection": {
      "Type": "AWS::EC2::VPCPeeringConnection",
      "Properties": {
        "VpcId": {
          "Ref": "vpc"
        },
        "PeerVpcId": {
          "Ref": "PeerVPCId"
        },
        "PeerOwnerId": {
          "Ref": "PeerVPCAccountId"
        },
        "PeerRoleArn": {
```

```

        "Ref": "PeerRoleArn"
      }
    }
  },
  "Outputs": {
    "VPCId": {
      "Value": {
        "Ref": "vpc"
      }
    },
    "VPCPeeringConnectionId": {
      "Value": {
        "Ref": "vpcPeeringConnection"
      }
    }
  }
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and a VPC Peering connection using the PeerRole to
  accept.
Parameters:
  PeerVPCAccountId:
    Type: String
  PeerVPCId:
    Type: String
  PeerRoleArn:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.2.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  vpcPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref vpc

```

```
    PeerVpcId: !Ref PeerVPCId
    PeerOwnerId: !Ref PeerVPCAccountId
    PeerRoleArn: !Ref PeerRoleArn
Outputs:
  VPCId:
    Value: !Ref vpc
  VPCPeeringConnectionId:
    Value: !Ref vpcPeeringConnection
```

5. Choisissez Suivant.
6. Attribuez un nom à la pile (par exemple, **VPC-peering-connection**).
7. Acceptez les valeurs par défaut, puis choisissez Suivant.
8. Choisissez Je reconnais que cela CloudFormation pourrait créer des ressources IAM, puis choisissez Create stack.

Créer un modèle avec une politique très restrictive

Vous pouvez souhaiter créer une politique très restrictive pour l'appairage de votre VPC avec un autre Compte AWS.

L'exemple de modèle suivant montre comment modifier le modèle de propriétaire d'appairage VPC (le compte de l'accepteur créé à l'étape 1 ci-dessus) afin qu'il soit plus restrictif.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  },
  "Resources": {
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Action": [
```

```

        "sts:AssumeRole"
    ],
    "Effect": "Allow",
    "Principal": {
        "AWS": {
            "Ref": "PeerRequesterAccountId"
        }
    }
}
],
},
"Path": "/",
"Policies": [
    {
        "PolicyDocument": {
            "Statement": [
                {
                    "Action": "ec2:acceptVpcPeeringConnection",
                    "Effect": "Allow",
                    "Resource": {
                        "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}"
                    }
                },
                {
                    "Action": "ec2:acceptVpcPeeringConnection",
                    "Condition": {
                        "StringEquals": {
                            "ec2:AcceptorVpc": {
                                "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}"
                            }
                        }
                    },
                    "Effect": "Allow",
                    "Resource": {
                        "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-
peering-connection/*"
                    }
                }
            ]
        },
        "Version": "2012-10-17"
    },
    "PolicyName": "root"
}

```



```

    ]
  }
},
"vpc":{
  "Type":"AWS::EC2::VPC",
  "Properties":{
    "CidrBlock":"10.1.0.0/16",
    "EnableDnsHostnames":false,
    "EnableDnsSupport":false,
    "InstanceTenancy":"default"
  }
}
},
"Outputs":{
  "RoleARN":{
    "Value":{
      "Fn::GetAtt":[
        "peerRole",
        "Arn"
      ]
    }
  },
  "VPCId":{
    "Value":{
      "Ref":"vpc"
    }
  }
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:

```

```

    - Action:
      - 'sts:AssumeRole'
    Effect: Allow
    Principal:
      AWS:
        Ref: PeerRequesterAccountId
  Path: /
  Policies:
    - PolicyDocument:
        Statement:
          - Action: 'ec2:acceptVpcPeeringConnection'
            Effect: Allow
            Resource:
              'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}'
          - Action: 'ec2:acceptVpcPeeringConnection'
            Condition:
              StringEquals:
                'ec2:AcceptorVpc':
                  'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}'
            Effect: Allow
            Resource:
              'Fn::Sub': >-
                arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-peering-
connection/*
        Version: 2012-10-17
        PolicyName: root
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16
      EnableDnsHostnames: false
      EnableDnsSupport: false
      InstanceTenancy: default
  Outputs:
    RoleARN:
      Value:
        'Fn::GetAtt':
          - peerRole
          - Arn
    VPCId:
      Value:
        Ref: vpc

```

Pour accéder au VPC, vous pouvez utiliser le même modèle de demandeur que celui de l'étape 2 ci-dessus.

Pour plus d'informations, consultez la section [Gestion des identités et des accès pour l'appairage VPC](#) dans le Guide de l'appairage Amazon VPC.

Effectuez des blue/green déploiements ECS en utilisant CodeDeploy CloudFormation

Pour mettre à jour une application exécutée sur Amazon Elastic Container Service (Amazon ECS), vous pouvez utiliser CodeDeploy une stratégie de déploiement bleu/vert. Cette stratégie permet de réduire au minimum les interruptions causées par le changement de version de l'application.

Lors d'un blue/green déploiement, vous créez un nouvel environnement d'application (appelé vert) en plus de votre environnement réel actuel (appelé bleu). Cela vous permet de surveiller et de tester l'environnement vert avant de rediriger le trafic en direct de l'environnement bleu vers l'environnement vert. Une fois que l'environnement vert dessert le trafic en direct, vous pouvez mettre fin à l'environnement bleu en toute sécurité.

Pour effectuer des déploiements CodeDeploy bleu/vert sur ECS à l'aide de CloudFormation, vous devez inclure les informations suivantes dans votre modèle de stack :

- Une section `Hooks` qui décrit un hook `AWS::CodeDeploy::BlueGreen`.
- Une section `Transform` qui spécifie la transformation `AWS::CodeDeployBlueGreen`.

Les rubriques suivantes vous guident dans la configuration d'un CloudFormation modèle pour un déploiement bleu/vert sur ECS.

Rubriques

- [À propos des blue/green déploiements](#)
- [Considérations relatives à la gestion des blue/green déploiements ECS à l'aide de CloudFormation](#)
- [Syntaxe du hook `AWS::CodeDeploy::BlueGreen`](#)
- [Exemple de modèle de déploiement bleu/vert](#)

À propos des blue/green déploiements

Cette rubrique fournit une vue d'ensemble du fonctionnement des blue/green déploiements avec CloudFormation. Il explique également comment préparer votre CloudFormation modèle pour les déploiements bleu/vert.

Rubriques

- [Comment ça marche](#)
- [Mises à jour de ressources déclenchant un déploiement vert](#)
- [Préparation de votre modèle pour effectuer des blue/green déploiements ECS](#)
- [Modélisation de votre blue/green déploiement à l'aide de CloudFormation ressources](#)
- [Jeux de modifications](#)
- [Surveillance des événements de pile](#)
- [Autorisations IAM pour les déploiements blue/green](#)

Comment ça marche

Lorsque vous l'utilisez CloudFormation pour effectuer blue/green des déploiements ECS CodeDeploy, vous commencez par créer un modèle de pile qui définit les ressources pour vos environnements d'applications bleus et verts, notamment en spécifiant les paramètres de routage et de stabilisation du trafic à utiliser. Ensuite, vous créez une pile à partir de ce modèle. Cela génère votre application bleue (actuelle). CloudFormation crée uniquement les ressources bleues lors de la création de la pile. Les ressources pour un déploiement vert ne sont pas créées tant qu'elles ne sont pas requises.

Ensuite, si, lors d'une future mise à jour de la pile, vous mettez à jour la définition des tâches ou les ressources de l'ensemble de tâches dans votre application CloudFormation bleue, procédez comme suit :

- Génère toutes les ressources d'environnement d'applications « vertes » nécessaires.
- Déplace le trafic en fonction des paramètres de routage de trafic spécifiés.
- Supprime les ressources « bleu »

Si une erreur se produit à un moment quelconque avant que le déploiement vert ne soit réussi et finalisé, CloudFormation rétablit la pile à son état initial avant que le déploiement vert entier ne soit lancé.

Mises à jour de ressources déclenchant un déploiement vert

Lorsque vous effectuez une mise à jour de la pile qui met à jour certaines propriétés de ressources ECS spécifiques, CloudFormation un processus de déploiement écologique est lancé. Les ressources qui déclenchent ce processus sont les suivantes :

- [AWS::ECS::TaskDefinition](#)
- [AWS::ECS::TaskSet](#)

Cependant, si les mises à jour apportées à ces ressources ne concernent pas des propriétés nécessitant un remplacement, aucun déploiement vert n'est déclenché. Pour de plus amples informations, veuillez consulter [Comprendre les comportements de mise à jour des ressources d'une pile](#).

Il est important de noter que vous ne pouvez pas combiner une mise à jour portant sur ces ressources avec des mises à jour d'autres ressources dans la même opération de mise à jour de pile. Si vous devez mettre à jour à la fois les ressources répertoriées et d'autres ressources dans la même pile, vous disposez de deux options :

- Effectuez deux opérations de mise à jour de pile distinctes : l'une qui inclut uniquement les mises à jour des ressources ci-dessus et l'autre qui inclut les modifications apportées à d'autres ressources.
- Supprimez les sections Hooks Transform de votre modèle, puis effectuez la mise à jour de la pile. Dans ce cas, CloudFormation n'effectuera pas de déploiement écologique.

Préparation de votre modèle pour effectuer des blue/green déploiements ECS

Pour activer blue/green les déploiements sur votre stack, incluez les sections suivantes dans votre modèle de stack avant d'effectuer une mise à jour de stack.

- Ajoutez une référence à la transformation AWS::CodeDeployBlueGreen dans votre modèle :

```
"Transform": [  
  "AWS::CodeDeployBlueGreen"  
],
```

- Ajoutez une section Hooks qui appelle le hook AWS::CodeDeploy::BlueGreen et spécifie les propriétés de votre déploiement. Pour de plus amples informations, veuillez consulter [Syntaxe du hook AWS::CodeDeploy::BlueGreen](#).

- Dans la section **Ressources**, définissez les ressources bleues et vertes pour votre déploiement.

Vous pouvez ajouter ces sections lorsque vous créez le modèle pour la première fois (c'est-à-dire avant de créer la pile elle-même), ou vous pouvez les ajouter à un modèle existant avant d'effectuer une mise à jour de la pile. Si vous spécifiez le blue/green déploiement d'une nouvelle pile, les ressources bleues CloudFormation ne sont créées que lors de la création de la pile. Les ressources destinées au déploiement vert ne sont créées que lorsqu'elles sont requises lors d'une mise à jour de la pile.

Modélisation de votre blue/green déploiement à l'aide de CloudFormation ressources

Pour effectuer un déploiement CodeDeploy bleu/vert sur ECS, votre CloudFormation modèle doit inclure les ressources qui modélisent votre déploiement, telles qu'un service Amazon ECS et un équilibreur de charge. Pour de plus amples informations sur ce que représentent ces ressources, veuillez consulter [Avant de commencer un déploiement Amazon ECS](#) dans le Guide de l'utilisateur AWS CodeDeploy .

Exigence	Ressource	Obligatoire/facultatif	Lance le blue/green déploiement en cas de remplacement ?
Cluster Amazon ECS	AWS::ECS::Cluster	Facultatif. Le cluster par défaut peut être utilisé.	Non
Amazon ECS service	AWS::ECS::Service	Obligatoire.	Non
Application Load Balancer ou Network Load Balancer	AWS::ECS::Service LoadBalancer	Obligatoire.	Non
Écouteur de production	AWS::ElasticLoadBalancingV2::Listener	Obligatoire.	Non
Écouteur de test	AWS::ElasticLoadBalancingV2::Listener	Facultatif.	Non

Exigence	Ressource	Obligatoire/facultatif	Lance le blue/green déploiement en cas de remplacement ?
Deux groupes cibles	AWS::ElasticLoadBalancingV2::TargetGroup	Obligatoire.	Non
Définition de tâche Amazon ECS	AWS::ECS::TaskDefinition	Obligatoire.	Oui
Conteneur de votre application Amazon ECS	AWS::ECS::TaskDefinition Container Definition Name	Obligatoire.	Non
Port de votre ensemble de tâches de remplacement	AWS::ECS::TaskDefinition PortMapping ContainerPort	Obligatoire.	Non

Jeux de modifications

Nous vous recommandons fortement de créer un jeu de modifications avant d'effectuer une mise à jour de pile qui initiera un déploiement vert. Cette option vous permet de voir les modifications réelles qui seront apportées à votre pile avant d'effectuer la mise à jour. Sachez que les modifications de ressources peuvent ne pas être répertoriées dans l'ordre dans lequel elles seront effectuées lors de la mise à jour de la pile. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

Surveillance des événements de pile

Vous pouvez afficher les événements de pile générés à chaque étape du déploiement ECS dans l'onglet Events (Événements) de la page Pile et à l'aide de l' AWS CLI. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Autorisations IAM pour les déploiements blue/green

CloudFormation Pour effectuer avec succès les déploiements bleu-vert, vous devez disposer des autorisations suivantes : CodeDeploy

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

Pour plus d'informations, consultez la rubrique [Actions, ressources et clés de condition pour CodeDeploy](#) dans la section Référence de l'autorisation de service.

Considérations relatives à la gestion des blue/green déploiements ECS à l'aide de CloudFormation

Le processus d'utilisation CloudFormation pour effectuer vos blue/green déploiements ECS CodeDeploy est différent d'un déploiement ECS standard utilisant uniquement CodeDeploy. Pour une compréhension détaillée de ces différences, consultez la section [Différences entre les blue/green déploiements Amazon ECS via CodeDeploy et CloudFormation](#) dans le guide de l'AWS CodeDeploy utilisateur.

Lorsque vous gérez votre blue/green déploiement à l'aide CloudFormation de certaines limites et considérations à prendre en compte :

- Seules les mises à jour de certaines ressources déclenchent un déploiement vert. Pour de plus amples informations, veuillez consulter [Mises à jour de ressources déclenchant un déploiement vert](#).
- Vous ne pouvez pas inclure, dans la même mise à jour de pile, à la fois des mises à jour de ressources qui déclenchent un déploiement vert et des mises à jour d'autres ressources. Pour de plus amples informations, veuillez consulter [Mises à jour de ressources déclenchant un déploiement vert](#).
- Vous ne pouvez spécifier qu'un seul service ECS comme cible de déploiement.
- Les paramètres dont les valeurs sont masquées ne CloudFormation peuvent pas être mis à jour CodeDeploy lors d'un déploiement écologique, ce qui entraînera une erreur et un échec de la mise à jour de la pile. Il s'agit des licences suivantes :
 - Paramètres définis avec l'attribut NoEcho.
 - Paramètres qui utilisent des références dynamiques pour récupérer leurs valeurs à partir de services externes. Pour plus d'informations sur les références dynamiques, consultez [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).
- Pour annuler un déploiement écologique toujours en cours, annulez la mise à jour de la pile dans CloudFormation CodeDeploy ou pas dans ECS. Pour de plus amples informations, veuillez consulter [Annuler la mise à jour d'une pile](#). Une fois qu'une mise à jour est terminée, vous ne

pouvez plus l'annuler. Vous pouvez toutefois relancer une mise à jour de pile avec les paramètres précédents.

- Les CloudFormation fonctionnalités suivantes ne sont actuellement pas prises en charge pour les modèles qui définissent les blue/green déploiements ECS :
 - Déclarer des [sorties](#) ou utiliser [Fn :: ImportValue](#) pour importer des valeurs depuis d'autres piles.
 - Importation de ressources. Pour plus d'informations sur l'importation de ressources, consultez [Importer AWS des ressources dans une CloudFormation pile](#).
 - Utilisation du hook `AWS::CodeDeploy::BlueGreen` dans un modèle contenant des piles imbriquées. Pour plus d'informations sur les piles imbriquées, consultez [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).
 - Utilisation du hook `AWS::CodeDeploy::BlueGreen` dans une pile imbriquée.

Syntaxe du hook **AWS::CodeDeploy::BlueGreen**

La syntaxe suivante décrit la structure d'un `AWS::CodeDeploy::BlueGreen` hook pour les blue/green déploiements ECS.

Syntaxe

```
"Hooks": {  
  "Logical ID": {  
    "Type": "AWS::CodeDeploy::BlueGreen",  
    "Properties": {  
      "TrafficRoutingConfig": {  
        "Type": "Traffic routing type",  
        "TimeBasedCanary": {  
          "StepPercentage": Integer,  
          "BakeTimeMins": Integer  
        },  
        "TimeBasedLinear": {  
          "StepPercentage": Integer,  
          "BakeTimeMins": Integer  
        }  
      },  
      "AdditionalOptions": {"TerminationWaitTimeInMinutes": Integer},  
      "LifecycleEventHooks": {  
        "BeforeInstall": "FunctionName",  
        "AfterInstall": "FunctionName",  
        "AfterAllowTestTraffic": "FunctionName",
```

```

    "BeforeAllowTraffic": "FunctionName",
    "AfterAllowTraffic": "FunctionName"
  },
  "ServiceRole": "CodeDeployServiceRoleName",
  "Applications": [
    {
      "Target": {
        "Type": "AWS::ECS::Service",
        "LogicalID": "Logical ID of AWS::ECS::Service"
      },
      "ECSAttributes": {
        "TaskDefinitions": [
          "Logical ID of AWS::ECS::TaskDefinition (Blue)",
          "Logical ID of AWS::ECS::TaskDefinition (Green)"
        ],
        "TaskSets": [
          "Logical ID of AWS::ECS::TaskSet (Blue)",
          "Logical ID of AWS::ECS::TaskSet (Green)"
        ],
        "TrafficRouting": {
          "ProdTrafficRoute": {
            "Type": "AWS::ElasticLoadBalancingV2::Listener",
            "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener  

(Production)"
          },
          "TestTrafficRoute": {
            "Type": "AWS::ElasticLoadBalancingV2::Listener",
            "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener  

(Test)"
          },
          "TargetGroups": [
            "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Blue)",
            "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Green)"
          ]
        }
      }
    }
  ]
}

```

Propriétés

ID logique (également appelé nom logique)

L'ID logique d'un hook déclaré dans la section Hooks du modèle. L'ID logique doit être alphanumérique (A-Za-z0-9) et ne doit correspondre à aucun autre ID logique dans le modèle.

Obligatoire : oui

Type

Type de hook. AWS::CodeDeploy::BlueGreen

Obligatoire : oui

Properties

Propriétés du hook.

Obligatoire : oui

TrafficRoutingConfig

Paramètres de configuration du routage du trafic.

Obligatoire : non

La configuration par défaut est le déplacement du trafic Canary basé sur le temps, avec un pourcentage d'étape de 15 % et un temps d'attente de cinq minutes.

Type

Type de déplacement de trafic utilisé par la configuration de déploiement.

Valeurs valides : AllAtOnce | TimeBasedCanary | TimeBasedLinear

Obligatoire : oui

TimeBasedCanary

Spécifie une configuration qui déplace le trafic d'une version du déploiement vers une autre en deux incréments.

Requis : sous certaines conditions. Si vous spécifiez TimeBasedCanary comme type de routage du trafic, vous devez inclure le paramètre TimeBasedCanary.

StepPercentage

Pourcentage de trafic à déplacer dans le premier incrément d'un déploiement TimeBasedCanary. Le pourcentage d'étape doit être de 14 % ou plus.

Obligatoire : non

BakeTimeMins

Nombre de minutes entre les premier et deuxième déplacements de trafic d'un déploiement TimeBasedCanary.

Obligatoire : non

TimeBasedLinear

Spécifie une configuration qui déplace le trafic d'une version du déploiement vers une autre par incréments égaux, avec un nombre égal de minutes entre chaque incrément.

Requis : sous certaines conditions. Si vous spécifiez TimeBasedLinear comme type de routage du trafic, vous devez inclure le paramètre TimeBasedLinear.

StepPercentage

Pourcentage de trafic déplacé au début de chaque incrément d'un déploiement TimeBasedLinear. Le pourcentage d'étape doit être de 14 % ou plus.

Obligatoire : non

BakeTimeMins

Nombre de minutes entre chaque déplacement de trafic incrémentiel d'un déploiement TimeBasedLinear.

Obligatoire : non

AdditionalOptions

Options supplémentaires pour le blue/green déploiement.

Obligatoire : non

TerminationWaitTimeInMinutes

Spécifie le temps d'attente, en minutes, avant de mettre fin aux ressources bleues.

Obligatoire : non

LifecycleEventHooks

Utilisez les hooks d'événements du cycle de vie pour spécifier une fonction Lambda CodeDeploy pouvant être appelée pour valider un déploiement. Vous pouvez utiliser la même fonction ou une autre fonction pour les événements du cycle de vie du déploiement. Une fois les tests de validation terminés, la `AfterAllowTraffic` fonction Lambda rappelle CodeDeploy et fournit un résultat de `Succeeded` ou `Failed`. Pour plus d'informations, consultez la [section « crochets AppSpec »](#) du guide de l'AWS CodeDeploy utilisateur.

Obligatoire : non

`BeforeInstall`

Fonction permettant d'exécuter des tâches avant que l'ensemble des tâches de remplacement ne soit créé.

Obligatoire : non

`AfterInstall`

Fonction permettant d'exécuter des tâches après que l'ensemble des tâches de remplacement a été créé et que l'un des groupes cibles lui a été associé.

Obligatoire : non

`AfterAllowTestTraffic`

Fonction permettant d'exécuter des tâches après que l'écouteur de test a dirigé le trafic vers l'ensemble des tâches de remplacement.

Obligatoire : non

`BeforeAllowTraffic`

Fonction permettant d'exécuter des tâches après l'association du second groupe cible avec l'ensemble des tâches de remplacement, mais avant que le trafic ne soit déplacé vers ce dernier.

Obligatoire : non

`AfterAllowTraffic`

Fonction permettant d'exécuter des tâches une fois que le second groupe cible a dirigé le trafic vers l'ensemble des tâches de remplacement.

Obligatoire : non

ServiceRole

Le rôle d'exécution CloudFormation à utiliser pour effectuer les déploiements bleu-vert. Pour la liste des autorisations nécessaires, consultez [Autorisations IAM pour les déploiements blue/green](#).

Obligatoire : non

Applications

Spécifie les propriétés de l'application Amazon ECS.

Obligatoire : oui

Target

Obligatoire : oui

Type

Type de la ressource.

Obligatoire : oui

LogicalID

ID logique de la ressource.

Obligatoire : oui

ECSAttributes

Ressources qui représentent les différentes exigences du déploiement de votre application Amazon ECS.

Obligatoire : oui

TaskDefinitions

ID logique de la ressource [AWS::ECS::TaskDefinition](#) pour exécuter le conteneur Docker qui contient votre application Amazon ECS.

Obligatoire : oui

TaskSets

Logique IDs des [AWS::ECS::TaskSet](#)ressources à utiliser comme ensembles de tâches pour l'application.

Obligatoire : oui

TrafficRouting

Spécifie les ressources utilisées pour le routage du trafic.

Obligatoire : oui

ProdTrafficRoute

Un écouteur est utilisé par votre équilibreur de charge pour diriger le trafic vers vos groupes cibles.

Obligatoire : oui

Type

Type de la ressource. `AWS::ElasticLoadBalancingV2::Listener`

Obligatoire : oui

LogicalID

ID logique de la ressource.

Obligatoire : oui

TestTrafficRoute

Un écouteur est utilisé par votre équilibreur de charge pour diriger le trafic vers vos groupes cibles.

Obligatoire : oui

Type

Type de la ressource. `AWS::ElasticLoadBalancingV2::Listener`

Obligatoire : oui

LogicalID

ID logique de la ressource.

Obligatoire : non

TargetGroups

ID logique des ressources à utiliser comme groupes cibles pour acheminer le trafic vers la cible enregistrée.

Obligatoire : oui

Exemple de modèle de déploiement bleu/vert

L'exemple de modèle suivant configure un déploiement CodeDeploy bleu/vert sur ECS, avec une progression du routage du trafic de 15 % par étape et une période de stabilisation de 5 minutes entre chaque étape.

La création d'une pile à partir de ce modèle configure la configuration initiale du déploiement. Si vous apportez ensuite des modifications aux propriétés de la BlueTaskSet ressource qui nécessitent son remplacement, vous CloudFormation lancerez un déploiement écologique dans le cadre de la mise à jour de la pile.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": { "Type": "AWS::EC2::VPC::Id" },
    "Subnet1": { "Type": "AWS::EC2::Subnet::Id" },
    "Subnet2": { "Type": "AWS::EC2::Subnet::Id" }
  },
  "Transform": [ "AWS::CodeDeployBlueGreen" ],
  "Hooks": {
    "CodeDeployBlueGreenHook": {
      "Type": "AWS::CodeDeploy::BlueGreen",
      "Properties": {
        "TrafficRoutingConfig": {
          "Type": "TimeBasedCanary",
          "TimeBasedCanary": {
            "StepPercentage": 15,
            "BakeTimeMins": 5
          }
        }
      },
    },
    "Applications": [
      {
```



```

    "Target":{
      "Type":"AWS::ECS::Service",
      "LogicalID":"ECSDemoService"
    },
    "ECSAttributes":{
      "TaskDefinitions":[ "BlueTaskDefinition","GreenTaskDefinition" ],
      "TaskSets":[ "BlueTaskSet","GreenTaskSet" ],
      "TrafficRouting":{
        "ProdTrafficRoute":{
          "Type":"AWS::ElasticLoadBalancingV2::Listener",
          "LogicalID":"ALBListenerProdTraffic"
        },
        "TargetGroups":[ "ALBTargetGroupBlue","ALBTargetGroupGreen" ]
      }
    }
  }
]
}
},
"Resources":{
  "ExampleSecurityGroup":{
    "Type":"AWS::EC2::SecurityGroup",
    "Properties":{
      "GroupDescription":"Security group for ec2 access",
      "VpcId":{"Ref":"Vpc" },
      "SecurityGroupIngress":[
        {
          "IpProtocol":"tcp",
          "FromPort":80,
          "ToPort":80,
          "CidrIp":"0.0.0.0/0"
        },
        {
          "IpProtocol":"tcp",
          "FromPort":8080,
          "ToPort":8080,
          "CidrIp":"0.0.0.0/0"
        },
        {
          "IpProtocol":"tcp",
          "FromPort":22,
          "ToPort":22,
          "CidrIp":"0.0.0.0/0"
        }
      ]
    }
  }
}

```

```

    }
  ]
}
},
"ALBTargetGroupBlue":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{ "HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{ "Ref":"Vpc" }
  }
},
"ALBTargetGroupGreen":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{ "HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{ "Ref":"Vpc" }
  }
},
"ExampleALB":{
  "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties":{
    "Scheme":"internet-facing",

```

```
    "SecurityGroups": [{ "Ref": "ExampleSecurityGroup" }],
    "Subnets": [{ "Ref": "Subnet1" }, { "Ref": "Subnet2" }],
    "Tags": [{ "Key": "Group", "Value": "Example" }],
    "Type": "application",
    "IpAddressType": "ipv4"
  }
},
"ALBListenerProdTraffic": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [
      {
        "Type": "forward",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": { "Ref": "ALBTargetGroupBlue" },
              "Weight": 1
            }
          ]
        }
      }
    ],
    "LoadBalancerArn": { "Ref": "ExampleALB" },
    "Port": 80,
    "Protocol": "HTTP"
  }
},
"ALBListenerProdRule": {
  "Type": "AWS::ElasticLoadBalancingV2::ListenerRule",
  "Properties": {
    "Actions": [
      {
        "Type": "forward",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": { "Ref": "ALBTargetGroupBlue" },
              "Weight": 1
            }
          ]
        }
      }
    ]
  }
},
],
```

```

    "Conditions": [
      {
        "Field": "http-header",
        "HTTPHeaderConfig": {
          "HTTPHeaderName": "User-Agent",
          "Values": [ "Mozilla" ]
        }
      }
    ],
    "ListenerArn": { "Ref": "ALBListenerProdTraffic" },
    "Priority": 1
  }
},
"ECSTaskExecutionRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  },
  "ManagedPolicyArns": [ "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" ]
},
"BlueTaskDefinition": {
  "Type": "AWS::ECS::TaskDefinition",
  "Properties": {
    "ExecutionRoleArn": {
      "Fn::GetAtt": [ "ECSTaskExecutionRole", "Arn" ]
    },
    "ContainerDefinitions": [
      {
        "Name": "DemoApp",
        "Image": "nginxdemos/hello:latest",
        "Essential": true,

```

```

        "PortMappings":[
            {
                "HostPort":80,
                "Protocol":"tcp",
                "ContainerPort":80
            }
        ]
    },
    "RequiresCompatibilities":[ "FARGATE" ],
    "NetworkMode":"awsvpc",
    "Cpu":"256",
    "Memory":"512",
    "Family":"ecs-demo"
}
},
"ECSDemoCluster":{
    "Type":"AWS::ECS::Cluster",
    "Properties":{}
},
"ECSDemoService":{
    "Type":"AWS::ECS::Service",
    "Properties":{
        "Cluster":{"Ref":"ECSDemoCluster" },
        "DesiredCount":1,
        "DeploymentController":{"Type":"EXTERNAL" }
    }
},
"BlueTaskSet":{
    "Type":"AWS::ECS::TaskSet",
    "Properties":{
        "Cluster":{"Ref":"ECSDemoCluster" },
        "LaunchType":"FARGATE",
        "NetworkConfiguration":{
            "AwsVpcConfiguration":{
                "AssignPublicIp":"ENABLED",
                "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
                "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }]
            }
        }
    },
    "PlatformVersion":"1.4.0",
    "Scale":{
        "Unit":"PERCENT",
        "Value":100
    }
}

```

```

    },
    "Service":{ "Ref":"ECSDemoService"},
    "TaskDefinition":{ "Ref":"BlueTaskDefinition" },
    "LoadBalancers":[
      {
        "ContainerName":"DemoApp",
        "ContainerPort":80,
        "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" }
      }
    ]
  },
  "PrimaryTaskSet":{
    "Type":"AWS::ECS::PrimaryTaskSet",
    "Properties":{
      "Cluster":{ "Ref":"ECSDemoCluster" },
      "Service":{ "Ref":"ECSDemoService" },
      "TaskSetId":{ "Fn::GetAtt":[ "BlueTaskSet","Id" ]
    }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  Vpc:
    Type: AWS::EC2::VPC::Id
  Subnet1:
    Type: AWS::EC2::Subnet::Id
  Subnet2:
    Type: AWS::EC2::Subnet::Id
Transform:
  - 'AWS::CodeDeployBlueGreen'
Hooks:
  CodeDeployBlueGreenHook:
    Type: AWS::CodeDeploy::BlueGreen
    Properties:
      TrafficRoutingConfig:
        Type: TimeBasedCanary
        TimeBasedCanary:

```

```
    StepPercentage: 15
    BakeTimeMins: 5
  Applications:
    - Target:
        Type: AWS::ECS::Service
        LogicalID: ECSDemoService
    ECSAttributes:
      TaskDefinitions:
        - BlueTaskDefinition
        - GreenTaskDefinition
      TaskSets:
        - BlueTaskSet
        - GreenTaskSet
      TrafficRouting:
        ProdTrafficRoute:
          Type: AWS::ElasticLoadBalancingV2::Listener
          LogicalID: ALBListenerProdTraffic
        TargetGroups:
          - ALBTargetGroupBlue
          - ALBTargetGroupGreen
```

Resources:**ExampleSecurityGroup:**

```
Type: AWS::EC2::SecurityGroup
```

Properties:

```
GroupDescription: Security group for ec2 access
```

```
VpcId: !Ref Vpc
```

SecurityGroupIngress:

- IpProtocol: tcp
FromPort: 80
ToPort: 80
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
FromPort: 8080
ToPort: 8080
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
FromPort: 22
ToPort: 22
CidrIp: 0.0.0.0/0

ALBTargetGroupBlue:

```
Type: AWS::ElasticLoadBalancingV2::TargetGroup
```

Properties:

```
HealthCheckIntervalSeconds: 5
```

```
HealthCheckPath: /
```

```
HealthCheckPort: '80'
HealthCheckProtocol: HTTP
HealthCheckTimeoutSeconds: 2
HealthyThresholdCount: 2
Matcher:
  HttpCode: '200'
Port: 80
Protocol: HTTP
Tags:
  - Key: Group
    Value: Example
TargetType: ip
UnhealthyThresholdCount: 4
VpcId: !Ref Vpc
ALBTargetGroupGreen:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 2
    HealthyThresholdCount: 2
    Matcher:
      HttpCode: '200'
    Port: 80
    Protocol: HTTP
    Tags:
      - Key: Group
        Value: Example
    TargetType: ip
    UnhealthyThresholdCount: 4
    VpcId: !Ref Vpc
ExampleALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    SecurityGroups:
      - !Ref ExampleSecurityGroup
    Subnets:
      - !Ref Subnet1
      - !Ref Subnet2
    Tags:
      - Key: Group
```



```
    Value: Example
    Type: application
    IpAddressType: ipv4
ALBListenerProdTraffic:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
      ForwardConfig:
        TargetGroups:
          - TargetGroupArn: !Ref ALBTargetGroupBlue
            Weight: 1
    LoadBalancerArn: !Ref ExampleALB
    Port: 80
    Protocol: HTTP
ALBListenerProdRule:
  Type: AWS::ElasticLoadBalancingV2::ListenerRule
  Properties:
    Actions:
      - Type: forward
      ForwardConfig:
        TargetGroups:
          - TargetGroupArn: !Ref ALBTargetGroupBlue
            Weight: 1
    Conditions:
      - Field: http-header
        HttpHeaderConfig:
          HttpHeaderName: User-Agent
          Values:
            - Mozilla
    ListenerArn: !Ref ALBListenerProdTraffic
    Priority: 1
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Sid: ''
          Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: 'sts:AssumeRole'
    ManagedPolicyArns:
```

```
- 'arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy'
BlueTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ExecutionRoleArn: !GetAtt
      - ECSTaskExecutionRole
      - Arn
    ContainerDefinitions:
      - Name: DemoApp
        Image: 'nginxdemos/hello:latest'
        Essential: true
        PortMappings:
          - HostPort: 80
            Protocol: tcp
            ContainerPort: 80
    RequiresCompatibilities:
      - FARGATE
    NetworkMode: awsvpc
    Cpu: '256'
    Memory: '512'
    Family: ecs-demo
ECSDemoCluster:
  Type: AWS::ECS::Cluster
  Properties: {}
ECSDemoService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref ECSDemoCluster
    DesiredCount: 1
    DeploymentController:
      Type: EXTERNAL
BlueTaskSet:
  Type: AWS::ECS::TaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
      SecurityGroups:
        - !Ref ExampleSecurityGroup
      Subnets:
        - !Ref Subnet1
        - !Ref Subnet2
```

```
PlatformVersion: 1.4.0
Scale:
  Unit: PERCENT
  Value: 100
Service: !Ref ECSDemoService
TaskDefinition: !Ref BlueTaskDefinition
LoadBalancers:
  - ContainerName: DemoApp
    ContainerPort: 80
    TargetGroupArn: !Ref ALBTargetGroupBlue
PrimaryTaskSet:
  Type: AWS::ECS::PrimaryTaskSet
Properties:
  Cluster: !Ref ECSDemoCluster
  Service: !Ref ECSDemoService
  TaskSetId: !GetAtt
    - BlueTaskSet
    - Id
```

CloudFormation extraits de modèles

Cette section fournit plusieurs exemples de scénarios que vous pouvez utiliser pour comprendre comment déclarer différents composants de modèle CloudFormation . Vous pouvez également utiliser ces extraits comme point de départ pour les sections de vos modèles personnalisés.

Note

Comme les CloudFormation modèles doivent être conformes au format JSON, aucun caractère de continuation de ligne n'est prévu. Le retour à la ligne des extraits dans ce document peut être aléatoire si la ligne dépasse 68 caractères.

Rubriques

- [Extraits de modèle généraux](#)
- [Extraits de CloudFormation modèles de mise à l'échelle automatique](#)
- [AWS Extraits du modèle de console de facturation](#)
- [CloudFormation extraits de modèles](#)
- [Extraits CloudFront de modèles Amazon](#)
- [Extraits CloudWatch de modèles Amazon](#)

- [Extraits du modèle Amazon CloudWatch Logs](#)
- [Extraits de modèle Amazon DynamoDB](#)
- [Extraits EC2 CloudFormation de modèles Amazon](#)
- [Modèles d'exemple Amazon Elastic Container Service](#)
- [Exemple de modèle Amazon Elastic File System](#)
- [Extraits de modèle Elastic Beanstalk](#)
- [Extraits de modèle Elastic Load Balancing](#)
- [Gestion des identités et des accès AWS extraits de modèles](#)
- [AWS Lambda modèle](#)
- [Extraits de modèle Amazon Redshift](#)
- [Extraits de modèle Amazon RDS](#)
- [Extraits de modèle Route 53](#)
- [Extraits de modèle Amazon S3](#)
- [Extraits de modèle Amazon SNS](#)
- [Extraits de modèle Amazon SQS](#)
- [Extraits de modèles Amazon Timestream](#)

Extraits de modèle généraux

Les exemples suivants présentent différentes fonctionnalités du CloudFormation modèle qui ne sont pas spécifiques à un AWS service.

Rubriques

- [Propriété UserData encodée en Base64](#)
- [Propriété UserData encodée en Base64 avec AccessKey et SecretKey](#)
- [Section Parameters avec un seul paramètre de type String](#)
- [Section Parameters avec le paramètre String et une contrainte d'expression régulière](#)
- [Section Parameters avec paramètre numérique et contraintes MinValue et MaxValue](#)
- [Parameterssection avec paramètre numérique avec AllowedValues contrainte](#)
- [Section Parameters avec un seul paramètre de type CommaDelimitedList](#)
- [Section Parameters avec une valeur de paramètre basée sur un pseudo-paramètre](#)
- [Section Mapping avec trois mappages](#)

- [Description basé sur une chaîne littérale](#)
- [Section Outputs avec une seule sortie de chaîne littérale](#)
- [Section Outputs avec une seule référence de ressource et une seule pseudo-référence](#)
- [Section Outputs avec une sortie basée sur une fonction, une chaîne littérale, une référence et un pseudo-paramètre](#)
- [Version de format de modèle](#)
- [propriétéAWS Tags](#)

Propriété UserData encodée en Base64

Cet exemple montre l'assemblage d'une propriété UserData à l'aide des fonctions Fn::Base64 et Fn::Join. Les références MyValue et MyName sont des paramètres qui doivent être définis dans la section Parameters du modèle. La chaîne littérale Hello World n'est qu'une autre valeur que cet exemple transmet dans le cadre du paramètre UserData.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ ",", [
      { "Ref" : "MyValue" },
      { "Ref" : "MyName" },
      "Hello World" ] ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    Ref: MyValue
    Ref: MyName
    Hello World
```

Propriété UserData encodée en Base64 avec AccessKey et SecretKey

Cet exemple montre l'assemblage d'une propriété UserData à l'aide des fonctions Fn::Base64 et Fn::Join. Il inclut les informations AccessKey et SecretKey. Les références AccessKey et SecretKey sont des paramètres qui doivent être définis dans la section Parameters du modèle.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ "", [
      "ACCESS_KEY=", { "Ref" : "AccessKey" },
      "SECRET_KEY=", { "Ref" : "SecretKey" } ]
    ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    ACCESS_KEY=${AccessKey}
    SECRET_KEY=${SecretKey}
```

Section Parameters avec un seul paramètre de type String

L'exemple suivant illustre la déclaration d'une section Parameters valide dans laquelle un seul paramètre de type String est déclaré.

JSON

```
"Parameters" : {
  "UserName" : {
    "Type" : "String",
    "Default" : "nonadmin",
    "Description" : "Assume a vanilla user if no command-line spec provided"
  }
}
```

YAML

```
Parameters:
  UserName:
    Type: String
    Default: nonadmin
    Description: Assume a vanilla user if no command-line spec provided
```

Section Parameters avec le paramètre String et une contrainte d'expression régulière

L'exemple suivant illustre la déclaration d'une section Parameters valide dans laquelle un seul paramètre de type String est déclaré. La valeur AdminUserAccount par défaut du paramètre est de admin. La valeur de ce paramètre doit comprendre entre 1 et 16 caractères alphabétiques ou numériques, mais doit commencer par un caractère alphabétique.

JSON

```
"Parameters" : {
  "AdminUserAccount": {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "The admin account user name",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "16",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
  }
}
```

YAML

```
Parameters:
  AdminUserAccount:
    Default: admin
    NoEcho: true
    Description: The admin account user name
    Type: String
    MinLength: 1
    MaxLength: 16
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
```

Section Parameters avec paramètre numérique et contraintes MinValue et MaxValue

L'exemple suivant illustre la déclaration d'une section Parameters valide dans laquelle un seul paramètre de type Number est déclaré. Le paramètre WebServerPort possède la valeur par défaut 80. Ses valeurs minimale et maximale s'élèvent respectivement à 1 et 65535.

JSON

```
"Parameters" : {
```

```
"WebServerPort": {
  "Default": "80",
  "Description" : "TCP/IP port for the web server",
  "Type": "Number",
  "MinValue": "1",
  "MaxValue": "65535"
}
```

YAML

```
Parameters:
  WebServerPort:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    MinValue: 1
    MaxValue: 65535
```

Parameterssection avec paramètre numérique avec AllowedValues contrainte

L'exemple suivant illustre la déclaration d'une section Parameters valide dans laquelle un seul paramètre de type Number est déclaré. Le paramètre WebServerPort possède la valeur par défaut 80. Seules les valeurs 80 et 8888 sont autorisées.

JSON

```
"Parameters" : {
  "WebServerPortLimited": {
    "Default": "80",
    "Description" : "TCP/IP port for the web server",
    "Type": "Number",
    "AllowedValues" : ["80", "8888"]
  }
}
```

YAML

```
Parameters:
  WebServerPortLimited:
    Default: 80
```



```
Description: TCP/IP port for the web server
Type: Number
AllowedValues:
- 80
- 8888
```

Section Parameters avec un seul paramètre de type CommaDelimitedList

L'exemple suivant illustre une déclaration de section `Parameters` valide dans laquelle un seul paramètre de type `CommaDelimitedList` est déclaré. La propriété `NoEcho` est définie sur `TRUE`, ce qui masquera sa valeur par des astérisques (`*****`) dans la sortie `describe-stacks`, à l'exception des informations stockées aux emplacements spécifiés ci-dessous.

Important

L'utilisation de l'attribut `NoEcho` ne masque aucune information stockée dans les lieux suivants :

- La section des `Metadata` modèles. CloudFormation ne transforme, ne modifie ni n'expurge aucune information que vous incluez dans `Metadata` cette section. Pour de plus amples informations, veuillez consulter [Métadonnées](#).
- La section de modèle `Outputs` Pour de plus amples informations, veuillez consulter [Sorties](#)
- L'attribut `Metadata` d'une définition de ressource. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Nous vous recommandons vivement de ne pas utiliser ces mécanismes pour inclure des informations sensibles, telles que des mots de passe ou des secrets.

Important

Plutôt que d'intégrer des informations sensibles directement dans vos CloudFormation modèles, nous vous recommandons d'utiliser des paramètres dynamiques dans le modèle de pile pour référencer les informations sensibles stockées et gérées à l'extérieur CloudFormation, par exemple dans le AWS Systems Manager Parameter Store ou AWS Secrets Manager.

Pour de plus amples informations, veuillez consulter la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

JSON

```
"Parameters" : {
  "UserRoles" : {
    "Type" : "CommaDelimitedList",
    "Default" : "guest,newhire",
    "NoEcho" : "TRUE"
  }
}
```

YAML

```
Parameters:
  UserRoles:
    Type: CommaDelimitedList
    Default: "guest,newhire"
    NoEcho: true
```

Section Parameters avec une valeur de paramètre basée sur un pseudo-paramètre

L'exemple suivant illustre les commandes qui utilisent les pseudo-paramètres `AWS::StackName` et `AWS::Region` dans les données utilisateur EC2. Pour plus d'informations sur les pseudo-paramètres, consultez [Obtenir des AWS valeurs à l'aide de pseudo-paramètres](#).

JSON

```
"UserData" : { "Fn::Base64" : { "Fn::Join" : ["", [
  "#!/bin/bash -xe\n",
  "yum install -y aws-cfn-bootstrap\n",

  "/opt/aws/bin/cfn-init -v ",
  "    --stack ", { "Ref" : "AWS::StackName" },
  "    --resource LaunchConfig ",
  "    --region ", { "Ref" : "AWS::Region" }, "\n",

  "/opt/aws/bin/cfn-signal -e $? ",
```

```

        "        --stack ", { "Ref" : "AWS::StackName" },
        "        --resource WebServerGroup ",
        "        --region ", { "Ref" : "AWS::Region" }, "\n"
    ]]]}
}

```

YAML

```

UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    yum update -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource LaunchConfig --
region ${AWS::Region}
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource WebServerGroup
--region ${AWS::Region}

```

Section Mapping avec trois mappages

L'exemple suivant illustre une déclaration de section Mapping valide qui contient trois mappages. Lorsque ce mappage est associé à une clé de mappage Stop, SlowDown, ou Go, il fournit les valeurs RVB assignées à l'attribut RGBColor correspondant.

JSON

```

"Mappings" : {
  "LightColor" : {
    "Stop" : {
      "Description" : "red",
      "RGBColor" : "RED 255 GREEN 0 BLUE 0"
    },
    "SlowDown" : {
      "Description" : "yellow",
      "RGBColor" : "RED 255 GREEN 255 BLUE 0"
    },
    "Go" : {
      "Description" : "green",
      "RGBColor" : "RED 0 GREEN 128 BLUE 0"
    }
  }
}

```

YAML

```
Mappings:
  LightColor:
    Stop:
      Description: red
      RGBColor: "RED 255 GREEN 0 BLUE 0"
    SlowDown:
      Description: yellow
      RGBColor: "RED 255 GREEN 255 BLUE 0"
    Go:
      Description: green
      RGBColor: "RED 0 GREEN 128 BLUE 0"
```

Description basé sur une chaîne littérale

L'exemple suivant illustre une déclaration de section `Description` valide dans laquelle la valeur est basée sur une chaîne littérale. Cet extrait peut s'appliquer à des modèles, des paramètres, des ressources, des propriétés ou des sorties.

JSON

```
"Description" : "Replace this value"
```

YAML

```
Description: "Replace this value"
```

Section Outputs avec une seule sortie de chaîne littérale

Cet exemple illustre une affectation de sortie basée sur une chaîne littérale.

JSON

```
"Outputs" : {
  "MyPhone" : {
    "Value" : "Please call 555-5555",
    "Description" : "A random message for aws cloudformation describe-stacks"
  }
}
```

YAML

```
Outputs:
  MyPhone:
    Value: Please call 555-5555
    Description: A random message for aws cloudformation describe-stacks
```

Section Outputs avec une seule référence de ressource et une seule pseudo-référence

Cet exemple illustre une section `Outputs` avec deux affectations de sortie. L'une est basée sur une ressource, et l'autre sur une pseudo-référence.

JSON

```
"Outputs" : {
  "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
  "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```

Section Outputs avec une sortie basée sur une fonction, une chaîne littérale, une référence et un pseudo-paramètre

Cet exemple illustre une section `Outputs` avec une seule affectation de sortie. La fonction `Join` est utilisée pour concaténer la valeur, à l'aide d'un signe pour cent comme délimiteur.

JSON

```
"Outputs" : {
  "MyOutput" : {
    "Value" : { "Fn::Join" :
      [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
    }
  }
}
```

```
    }  
  }  
}
```

YAML

```
Outputs:  
  MyOutput:  
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ] ]
```

Version de format de modèle

L'extrait suivant illustre une déclaration de section `AWSTemplateFormatVersion` valide.

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

propriétéAWS Tags

Cet exemple montre une AWS Tags propriété. Vous devez spécifier cette propriété dans la section Propriétés d'une ressource. Lorsque la ressource est créée, les balises que vous déclarez lui sont ajoutées.

JSON

```
"Tags" : [  
  {  
    "Key" : "keyname1",  
    "Value" : "value1"  
  },  
  {  
    "Key" : "keyname2",  
    "Value" : "value2"  
  }  
]
```

YAML

```
Tags:
-
  Key: "keyname1"
  Value: "value1"
-
  Key: "keyname2"
  Value: "value2"
```

Extraits de CloudFormation modèles de mise à l'échelle automatique

Avec Amazon EC2 Auto Scaling, vous pouvez redimensionner automatiquement EC2 les instances Amazon, soit avec des politiques de dimensionnement, soit avec un dimensionnement planifié. Les groupes Auto Scaling sont des ensembles d' EC2 instances Amazon qui activent des fonctionnalités de dimensionnement et de gestion de flotte automatiques, telles que les politiques de dimensionnement, les actions planifiées, les bilans de santé, les crochets du cycle de vie et l'équilibrage de charge.

Application Auto Scaling fournit un dimensionnement automatique de différentes ressources au-delà d'Amazon EC2, soit avec des politiques de dimensionnement, soit avec un dimensionnement planifié.

Vous pouvez créer et configurer des groupes Auto Scaling, des politiques de dimensionnement, des actions planifiées et d'autres ressources Auto Scaling dans le cadre de votre infrastructure à l'aide de CloudFormation modèles. Les modèles facilitent la gestion et l'automatisation du déploiement des ressources à autoscaling de manière répétable et cohérente.

Les exemples d'extraits de modèle suivants décrivent CloudFormation des ressources ou des composants pour Amazon EC2 Auto Scaling et Application Auto Scaling. Ces extraits sont conçus pour être intégrés dans un modèle et ne sont pas destinés à être exécutés indépendamment.

Catégories d'extraits

- [Configurez les ressources Amazon EC2 Auto Scaling avec CloudFormation](#)
- [Configurer les ressources Application Auto Scaling avec CloudFormation](#)

Configurez les ressources Amazon EC2 Auto Scaling avec CloudFormation

Les exemples suivants montrent différents extraits à inclure dans les modèles à utiliser avec Amazon EC2 Auto Scaling.

Catégories d'extraits

- [Créer un groupe Auto Scaling à instance unique](#)
- [Créer un groupe Auto Scaling avec un équilibreur de charge associé](#)
- [Créer un groupe Auto Scaling avec notifications](#)
- [Créer un groupe Auto Scaling qui utilise un CreationPolicy et un UpdatePolicy](#)
- [Création d'une stratégie de mise à l'échelle par étapes](#)
- [Exemples de groupes d'instances mixtes](#)
- [Exemples de configuration de lancement](#)

Créer un groupe Auto Scaling à instance unique

Cet exemple montre une ressource [AWS::AutoScaling::AutoScalingGroup](#) avec une seule instance pour vous aider à démarrer. La propriété `VPCZoneIdentifier` du groupe Auto Scaling spécifie une liste de sous-réseaux existants dans trois zones de disponibilité différentes. Vous devez spécifier le sous-réseau applicable IDs depuis votre compte avant de créer votre stack. La propriété `LaunchTemplate` fait référence à une ressource [AWS::EC2::LaunchTemplate](#) dont le nom logique `myLaunchTemplate` est défini ailleurs dans votre modèle.

Note

Pour des exemples de modèles de lancement, consultez [Créez des modèles de lancement avec CloudFormation](#) dans la section Extraits Amazon EC2 et la section [Exemples](#) dans la ressource `AWS::EC2::LaunchTemplate`.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
      }
    }
  }
},
```



```

    "Version" : {
      "Fn::GetAtt" : [
        "myLaunchTemplate",
        "LatestVersionNumber"
      ]
    },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '1'
    MinSize: '1'

```

Créer un groupe Auto Scaling avec un équilibreur de charge associé

Cet exemple montre une ressource [AWS::AutoScaling::AutoScalingGroup](#) pour l'équilibrage de charge sur plusieurs serveurs. Il spécifie les noms logiques des AWS ressources déclarées ailleurs dans le même modèle.

1. La propriété `VPCZoneIdentifier` spécifie les noms logiques de deux ressources [AWS::EC2::Subnet](#) où les instances EC2 du groupe Auto Scaling seront créées : `myPublicSubnet1` et `myPublicSubnet2`.
2. La propriété `LaunchTemplate` spécifie une ressource [AWS::EC2::LaunchTemplate](#) avec le nom logique `myLaunchTemplate`.
3. La propriété `TargetGroupARNs` répertorie les groupes cibles d'un Application Load Balancer ou Network Load Balancer utilisés pour acheminer le trafic vers le

groupe Auto Scaling. Dans cet exemple, un groupe cible est spécifié, une ressource [AWS::ElasticLoadBalancingV2::TargetGroup](#) portant le nom logique myTargetGroup.

JSON

```
"myServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ { "Ref" : "myPublicSubnet1" }, { "Ref" :
"myPublicSubnet2" } ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
      },
      "Version" : {
        "Fn::GetAtt" : [
          "myLaunchTemplate",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "5",
    "MinSize" : "1",
    "TargetGroupARNs" : [ { "Ref" : "myTargetGroup" } ]
  }
}
```

YAML

```
myServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref myPublicSubnet1
      - !Ref myPublicSubnet2
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    TargetGroupARNs:
```

- !Ref *myTargetGroup*

Consultez aussi

Pour un exemple détaillé qui crée un groupe Auto Scaling avec une politique de suivi des objectifs et d'échelonnement basée sur la métrique prédéfinie `ALBRequestCountPerTarget` pour votre Application Load Balancer, veuillez consulter la rubrique [Exemples](#) de la ressource `AWS::AutoScaling::ScalingPolicy`.

Créer un groupe Auto Scaling avec notifications

Cet exemple montre une ressource [AWS::AutoScaling::AutoScalingGroup](#) qui envoie des notifications Amazon SNS lorsque les événements spécifiés se produisent. La `NotificationConfigurations` propriété spécifie la rubrique SNS dans laquelle CloudFormation une notification est envoyée et les événements qui entraîneront l'envoi CloudFormation de notifications. Lorsque les événements spécifiés par `NotificationTypes` se produisent, CloudFormation envoie une notification à la rubrique SNS spécifiée par `TopicARN`. Lorsque vous lancez la pile, elle CloudFormation crée une [AWS::SNS::Subscription](#) ressource (`snsTopicForAutoScalingGroup`) déclarée dans le même modèle.

La propriété `VPCZoneIdentifier` du groupe Auto Scaling spécifie une liste de sous-réseaux existants dans trois zones de disponibilité différentes. Vous devez spécifier le sous-réseau applicable IDs depuis votre compte avant de créer votre stack. La propriété `LaunchTemplate` fait référence au nom logique d'une ressource [AWS::EC2::LaunchTemplate](#) déclarée ailleurs dans le même modèle.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": [
    "snsTopicForAutoScalingGroup"
  ],
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "logicalName"
      },
    },
    "Version" : {
      "Fn::GetAtt" : [
```

```

        "logicalName",
        "LatestVersionNumber"
    ]
}
},
"MaxSize" : "5",
"MinSize" : "1",
"NotificationConfigurations" : [
{
    "TopicARN" : { "Ref" : "snsTopicForAutoScalingGroup" },
    "NotificationTypes" : [
        "autoscaling:EC2_INSTANCE_LAUNCH",
        "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
        "autoscaling:EC2_INSTANCE_TERMINATE",
        "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
        "autoscaling:TEST_NOTIFICATION"
    ]
}
]
}
}
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - snsTopicForAutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref logicalName
      Version: !GetAtt logicalName.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref snsTopicForAutoScalingGroup
        NotificationTypes:
          - autoscaling:EC2_INSTANCE_LAUNCH
          - autoscaling:EC2_INSTANCE_LAUNCH_ERROR

```

- autoscaling:EC2_INSTANCE_TERMINATE
- autoscaling:EC2_INSTANCE_TERMINATE_ERROR
- autoscaling:TEST_NOTIFICATION

Créer un groupe Auto Scaling qui utilise un **CreationPolicy** et un **UpdatePolicy**

L'exemple suivant montre comment ajouter des attributs **CreationPolicy** et **UpdatePolicy** à une ressource [AWS::AutoScaling::AutoScalingGroup](#).

La politique de création d'échantillons empêche le groupe Auto Scaling d'atteindre le **CREATE_COMPLETE** statut tant qu'il n'a Count pas CloudFormation reçu un certain nombre de signaux de réussite lorsque le groupe est prêt. Pour signaler que le groupe Auto Scaling est prêt, un script d'assistant `cfn-signal` ajouté aux données utilisateur du modèle de lancement (non affiché) est exécuté sur les instances. Si les instances n'envoient pas de signal dans les limites spécifiées **Timeout**, cela CloudFormation suppose que les instances n'ont pas été créées, que la création de ressources échoue et CloudFormation annule la pile.

L'exemple de politique de mise à jour indique d'effectuer une mise CloudFormation à jour continue à l'aide de la **AutoScalingRollingUpdate** propriété. La mise à jour continue modifie le groupe Auto Scaling par petits lots (dans cet exemple, instance par instance) en fonction de **MaxBatchSize** et d'un temps d'interruption entre les lots de mises à jour basé sur **PauseTime**. L'**MinInstancesInService** attribut indique le nombre minimum d'instances qui doivent être en service au sein du groupe Auto Scaling lors de la mise CloudFormation à jour des anciennes instances.

L'attribut **WaitOnResourceSignals** est défini sur **true**. CloudFormation doit recevoir un signal de chaque nouvelle instance dans le délai **PauseTime** spécifié pour poursuivre la mise à jour. Lorsque la pile est en cours de mise à jour, les processus EC2 Auto Scaling suivants sont suspendus : **HealthCheck**, **ReplaceUnhealthy**, **AZRebalance**, **AlarmNotification** et **ScheduledActions**. Remarque : ne suspendez pas les types de processus **Launch**, **Terminate** ou **AddToLoadBalancer** (si le groupe Auto Scaling est utilisé avec Elastic Load Balancing) car cela peut empêcher le fonctionnement correct de la mise à jour continue.

La propriété **VPCZoneIdentifier** du groupe Auto Scaling spécifie une liste de sous-réseaux existants dans trois zones de disponibilité différentes. Vous devez spécifier le sous-réseau applicable IDs depuis votre compte avant de créer votre stack. La propriété **LaunchTemplate** fait référence au nom logique d'une ressource [AWS::EC2::LaunchTemplate](#) déclarée ailleurs dans le même modèle.

Pour plus d'informations sur les attributs `CreationPolicy` et `UpdatePolicy`, consultez la [référence des attributs de ressource](#).

JSON

```
{
  "Resources":{
    "myASG":{
      "CreationPolicy":{
        "ResourceSignal":{
          "Count":"3",
          "Timeout":"PT15M"
        }
      },
      "UpdatePolicy":{
        "AutoScalingRollingUpdate":{
          "MinInstancesInService":"3",
          "MaxBatchSize":"1",
          "PauseTime":"PT12M5S",
          "WaitOnResourceSignals":"true",
          "SuspendProcesses":[
            "HealthCheck",
            "ReplaceUnhealthy",
            "AZRebalance",
            "AlarmNotification",
            "ScheduledActions",
            "InstanceRefresh"
          ]
        }
      },
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "VPCZoneIdentifier":[ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
        "LaunchTemplate":{
          "LaunchTemplateId":{
            "Ref":"logicalName"
          },
          "Version":{
            "Fn::GetAtt":[
              "logicalName",
              "LatestVersionNumber"
            ]
          }
        }
      }
    },
  },
}
```

```
        "MaxSize": "5",
        "MinSize": "3"
    }
}
}
```

YAML

```
---
Resources:
  myASG:
    CreationPolicy:
      ResourceSignal:
        Count: '3'
        Timeout: PT15M
    UpdatePolicy:
      AutoScalingRollingUpdate:
        MinInstancesInService: '3'
        MaxBatchSize: '1'
        PauseTime: PT12M5S
        WaitOnResourceSignals: true
      SuspendProcesses:
        - HealthCheck
        - ReplaceUnhealthy
        - AZRebalance
        - AlarmNotification
        - ScheduledActions
        - InstanceRefresh
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
        - subnetIdAz2
        - subnetIdAz3
      LaunchTemplate:
        LaunchTemplateId: !Ref logicalName
        Version: !GetAtt logicalName.LatestVersionNumber
      MaxSize: '5'
      MinSize: '3'
```

Création d'une stratégie de mise à l'échelle par étapes

Cet exemple montre une ressource [AWS::AutoScaling::ScalingPolicy](#) qui étend le groupe Auto Scaling à l'aide d'une stratégie de mise à l'échelle par paliers. La propriété `AdjustmentType` spécifie `ChangeInCapacity`, ce qui signifie que `ScalingAdjustment` représente le nombre d'instances à ajouter (si `ScalingAdjustment` est positif) ou à supprimer (s'il est négatif). Dans cet exemple, le paramètre `ScalingAdjustment` indique 1. Par conséquent, la politique ajoute une instance EC2 dans le groupe lorsque le seuil de l'alarme est dépassé.

La ressource [AWS::CloudWatch::Alarm](#) `CPUAlarmHigh` spécifie la politique de mise à l'échelle `ASGScalingPolicyHigh` comme action à exécuter lorsque l'alarme est dans un état `ALARM` (`AlarmActions`). La propriété `Dimensions` fait référence au nom logique d'une ressource [AWS::AutoScaling::AutoScalingGroup](#) déclarée ailleurs dans le même modèle.

JSON

```
{
  "Resources":{
    "ASGScalingPolicyHigh":{
      "Type":"AWS::AutoScaling::ScalingPolicy",
      "Properties":{
        "AutoScalingGroupName":{"Ref":"logicalName" },
        "PolicyType":"StepScaling",
        "AdjustmentType":"ChangeInCapacity",
        "StepAdjustments":[
          {
            "MetricIntervalLowerBound":0,
            "ScalingAdjustment":1
          }
        ]
      }
    },
    "CPUAlarmHigh":{
      "Type":"AWS::CloudWatch::Alarm",
      "Properties":{
        "EvaluationPeriods":"2",
        "Statistic":"Average",
        "Threshold":"90",
        "AlarmDescription":"Scale out if CPU > 90% for 2 minutes",
        "Period":"60",
        "AlarmActions":[ { "Ref":"ASGScalingPolicyHigh" } ],

```



```

    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "AutoScalingGroupName",
        "Value": { "Ref": "logicalName" }
      }
    ],
    "ComparisonOperator": "GreaterThanThreshold",
    "MetricName": "CPUUtilization"
  }
}
}
}

```

YAML

```

---
Resources:
  ASGScalingPolicyHigh:
    Type: AWS::AutoScaling::ScalingPolicy
    Properties:
      AutoScalingGroupName: !Ref logicalName
      PolicyType: StepScaling
      AdjustmentType: ChangeInCapacity
      StepAdjustments:
        - MetricIntervalLowerBound: 0
          ScalingAdjustment: 1
  CPUAlarmHigh:
    Type: AWS::CloudWatch::Alarm
    Properties:
      EvaluationPeriods: 2
      Statistic: Average
      Threshold: 90
      AlarmDescription: 'Scale out if CPU > 90% for 2 minutes'
      Period: 60
      AlarmActions:
        - !Ref ASGScalingPolicyHigh
      Namespace: AWS/EC2
      Dimensions:
        - Name: AutoScalingGroupName
          Value:
            !Ref logicalName
      ComparisonOperator: GreaterThanThreshold

```

MetricName: CPUUtilization

Consultez aussi

Pour plus d'exemples de modèles de politiques de mise à l'échelle, veuillez consulter la rubrique [Exemples](#) dans la ressource `AWS::AutoScaling::ScalingPolicy`.

Exemples de groupes d'instances mixtes

Création d'un groupe Auto Scaling en utilisant la sélection du type d'instance basée sur des attributs

Cet exemple montre une ressource [AWS::AutoScaling::AutoScalingGroup](#) qui contient les informations nécessaires pour lancer un groupe d'instances mixtes à l'aide de la sélection de type d'instance basée sur les attributs. Vous spécifiez les valeurs minimales et maximales pour la propriété `VCpuCount` et la valeur minimale pour la propriété `MemoryMiB`. Tous les types d'instance utilisés par le groupe Auto Scaling doivent correspondre à vos attributs d'instance requis.

La propriété `VPCZoneIdentifier` du groupe Auto Scaling spécifie une liste de sous-réseaux existants dans trois zones de disponibilité différentes. Vous devez spécifier le sous-réseau applicable IDs depuis votre compte avant de créer votre stack. La propriété `LaunchTemplate` fait référence au nom logique d'une ressource [AWS::EC2::LaunchTemplate](#) déclarée ailleurs dans le même modèle.

JSON

```
{
  "Resources":{
    "myASG":{
      "Type":"AWS::AutoScaling::AutoScalingGroup",
      "Properties":{
        "VPCZoneIdentifier":[
          "subnetIdAz1",
          "subnetIdAz2",
          "subnetIdAz3"
        ],
        "MixedInstancesPolicy":{
          "LaunchTemplate":{
            "LaunchTemplateSpecification":{
              "LaunchTemplateId":{
                "Ref":"logicalName"
              },
              "Version":{
```

```

        "Fn::GetAtt": [
            "logicalName",
            "LatestVersionNumber"
        ]
    },
    "Overrides": [
        {
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 2048
                }
            }
        }
    ]
},
"MaxSize": "5",
"MinSize": "1"
}
}
}
}
}

```

YAML

```

---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
        - subnetIdAz2
        - subnetIdAz3
      MixedInstancesPolicy:
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref logicalName

```

```

    Version: !GetAtt logicalName.LatestVersionNumber
  Overrides:
    - InstanceRequirements:
        VCpuCount:
          Min: 2
          Max: 4
        MemoryMiB:
          Min: 2048
  MaxSize: '5'
  MinSize: '1'

```

Exemples de configuration de lancement

Créez une configuration de lancement

Cet exemple montre une ressource [AWS::AutoScaling::LaunchConfiguration](#) pour un groupe Auto Scaling dans lequel vous spécifiez des valeurs pour les propriétés ImageId, InstanceType et SecurityGroups. La propriété SecurityGroups spécifie à la fois le nom logique d'une ressource [AWS::EC2::SecurityGroup](#) spécifiée ailleurs dans le modèle et un groupe de sécurité EC2 existant nommé myExistingEC2SecurityGroup.

JSON

```

"mySimpleConfig" : {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
  "Properties" : {
    "ImageId" : "ami-02354e95b3example",
    "InstanceType" : "t3.micro",
    "SecurityGroups" : [ { "Ref" : "logicalName" }, "myExistingEC2SecurityGroup" ]
  }
}

```

YAML

```

mySimpleConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: ami-02354e95b3example
    InstanceType: t3.micro
    SecurityGroups:

```

- !Ref *logicalName*
- *myExistingEC2SecurityGroup*

Créer un groupe Auto Scaling qui utilise une configuration de lancement

Cet exemple montre une ressource [AWS::AutoScaling::AutoScalingGroup](#) avec une seule instance. La propriété `VPCZoneIdentifier` du groupe Auto Scaling spécifie une liste de sous-réseaux existants dans trois zones de disponibilité différentes. Vous devez spécifier le sous-réseau applicable IDs depuis votre compte avant de créer votre stack. La propriété `LaunchConfigurationName` fait référence à une ressource [AWS::AutoScaling::LaunchConfiguration](#) dont le nom logique `mySimpleConfig` est défini dans votre modèle.

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchConfigurationName" : { "Ref" : "mySimpleConfig" },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchConfigurationName: !Ref mySimpleConfig
    MaxSize: '1'
    MinSize: '1'
```

Configurer les ressources Application Auto Scaling avec CloudFormation

Cette section fournit des exemples de CloudFormation modèles pour les politiques de dimensionnement d'Application Auto Scaling et les actions planifiées pour différentes AWS ressources.

Important

Lorsqu'un extrait Application Auto Scaling est inclus dans le modèle, vous devez peut-être déclarer une dépendance sur la ressource évolutive spécifique créée via le modèle à l'aide de l'attribut `DependsOn`. Cela remplace le parallélisme par défaut et indique à CloudFormation d'opérer sur les ressources dans un ordre spécifié. Sinon, la configuration de mise à l'échelle peut être appliquée avant que la ressource ait été complètement configurée. Pour plus d'informations, voir la section consacrée à l'[Attribut DependsOn](#).

Catégories d'extraits

- [Création d'une politique de dimensionnement pour une AppStream flotte](#)
- [Créer une stratégie de mise à l'échelle pour un cluster de base de données Aurora](#)
- [Créer une politique de mise à l'échelle pour une table DynamoDB](#)
- [Créer une stratégie de mise à l'échelle pour un service Amazon ECS \(métriques : CPU et mémoire moyens\)](#)
- [Créer une politique de mise à l'échelle pour un service Amazon ECS \(métrique : nombre moyen de requêtes par cible\)](#)
- [Créer une action planifiée avec une expression cron pour une fonction Lambda](#)
- [Créer une action planifiée avec une expression at pour un parc d'instances Spot](#)

Création d'une politique de dimensionnement pour une AppStream flotte

Cet extrait montre comment créer une stratégie et l'appliquer à une ressource

[AWS::AppStream::Fleet](#) à l'aide de la ressource

[AWS::ApplicationAutoScaling::ScalingPolicy](#). La ressource

[AWS::ApplicationAutoScaling::ScalableTarget](#) déclare une cible évolutive à laquelle cette politique est appliquée. Application Auto Scaling peut mettre à l'échelle le nombre d'instances de parc avec un minimum d'une instance et un maximum de 20 instances. La politique maintient l'utilisation

moyenne de la capacité de la flotte à 75 %, avec des temps de stabilisation de montée en puissance et de mise à l'échelle horizontale de 300 secondes (5 minutes).

Il utilise les fonctions intrinsèques `Fn::Join` et `Rev` pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::AppStream::Fleet` spécifiée dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 20,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam:${AWS::AccountId}:role/
aws-service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet" },
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "fleet",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    },
    "ScalingPolicyAppStreamFleet" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu75" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
```

```
    "/",  
    [  
      "fleet",  
      {  
        "Ref" : "logicalName"  
      }  
    ]  
  ],  
  "TargetTrackingScalingPolicyConfiguration" : {  
    "TargetValue" : 75,  
    "PredefinedMetricSpecification" : {  
      "PredefinedMetricType" : "AppStreamAverageCapacityUtilization"  
    },  
    "ScaleInCooldown" : 300,  
    "ScaleOutCooldown" : 300  
  }  
}  
  
}
```

YAML

```

---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 20
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet'
      ServiceNamespace: appstream
      ScalableDimension: appstream:fleet:DesiredCapacity
      ResourceId: !Join
        - /
        - - fleet
          - !Ref logicalName
    ScalingPolicyAppStreamFleet:
      Type: AWS::ApplicationAutoScaling::ScalingPolicy

```



```

Properties:
  PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu75
  PolicyType: TargetTrackingScaling
  ServiceNamespace: appstream
  ScalableDimension: appstream:fleet:DesiredCapacity
  ResourceId: !Join
    - /
    - - fleet
      - !Ref logicalName
  TargetTrackingScalingPolicyConfiguration:
    TargetValue: 75
    PredefinedMetricSpecification:
      PredefinedMetricType: AppStreamAverageCapacityUtilization
    ScaleInCooldown: 300
    ScaleOutCooldown: 300

```

Créer une stratégie de mise à l'échelle pour un cluster de base de données Aurora

Dans cet extrait, vous enregistrez une ressource [AWS::RDS::DBCluster](#). La ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) indique que le cluster de bases de données doit être dimensionné de manière dynamique pour comporter entre une et huit répliques Aurora. Vous appliquez également une stratégie de mise à l'échelle du suivi des cibles au cluster à l'aide de la ressource [AWS::ApplicationAutoScaling::ScalingPolicy](#).

Dans cette configuration, la métrique prédéfinie `RDSReaderAverageCPUUtilization` est utilisée pour ajuster un cluster de bases de données Aurora en fonction d'une utilisation moyenne de l'UC de 40 % sur tous les répliques Aurora de ce cluster de bases de données Aurora. La configuration indique un temps de stabilisation de diminution en charge de 10 minutes et un temps de stabilisation de montée en charge de 5 minutes.

Cet exemple utilise la fonction intrinsèque `Fn::Sub` pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::RDS::DBCluster` spécifiée dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

JSON

```

{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 8,

```

```

        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/rds.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_RDSCluster" },
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" }
    }
},
"ScalingPolicyDBCluster" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu40" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" },
        "TargetTrackingScalingPolicyConfiguration" : {
            "TargetValue" : 40,
            "PredefinedMetricSpecification" : {
                "PredefinedMetricType" : "RDSReaderAverageCPUUtilization"
            },
            "ScaleInCooldown" : 600,
            "ScaleOutCooldown" : 300
        }
    }
}
}
}
}
}
}

```

YAML

```

---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 8
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/rds.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster'
      ServiceNamespace: rds

```

```
ScalableDimension: rds:cluster:ReadReplicaCount
ResourceId: !Sub cluster:${logicalName}
ScalingPolicyDBCluster:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu40
    PolicyType: TargetTrackingScaling
    ServiceNamespace: rds
    ScalableDimension: rds:cluster:ReadReplicaCount
    ResourceId: !Sub cluster:${logicalName}
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 40
      PredefinedMetricSpecification:
        PredefinedMetricType: RDSReaderAverageCPUUtilization
      ScaleInCooldown: 600
      ScaleOutCooldown: 300
```

Créer une politique de mise à l'échelle pour une table DynamoDB

Cet extrait montre comment créer une stratégie avec le type de stratégie `TargetTrackingScaling` et l'appliquer à une ressource [AWS::DynamoDB::Table](#) à l'aide de la ressource [AWS::ApplicationAutoScaling::ScalingPolicy](#). La ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) déclare une cible évolutive à laquelle cette politique est appliquée, avec un minimum de cinq unités de capacité d'écriture et un maximum de 15. La politique de mise à l'échelle fait évoluer la capacité d'écriture de la table pour maintenir une utilisation cible à 50 % par rapport à la métrique prédéfinie `DynamoDBWriteCapacityUtilization`.

Il utilise les fonctions intrinsèques `Fn::Join` et `Ref` pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::DynamoDB::Table` spécifiée dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

Note

Pour plus d'informations sur la création d'un CloudFormation modèle pour les ressources DynamoDB, consultez le billet de [blog Comment utiliser pour configurer le dimensionnement automatique CloudFormation pour les tables et les index Amazon DynamoDB sur le blog de base](#) de données. AWS

JSON

```

{
  "Resources" : {
    "WriteCapacityScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 15,
        "MinCapacity" : 5,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
        "ServiceNamespace" : "dynamodb",
        "ScalableDimension" : "dynamodb:table:WriteCapacityUnits",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "table",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    },
    "WriteScalingPolicy" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : "WriteScalingPolicy",
        "PolicyType" : "TargetTrackingScaling",
        "ScalingTargetId" : { "Ref" : "WriteCapacityScalableTarget" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 50.0,
          "ScaleInCooldown" : 60,
          "ScaleOutCooldown" : 60,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "DynamoDBWriteCapacityUtilization"
          }
        }
      }
    }
  }
}

```

```
}
```

YAML

```
---
Resources:
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ServiceNamespace: dynamodb
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ResourceId: !Join
        - /
        - - table
          - !Ref logicalName
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteScalingPolicy
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0
        ScaleInCooldown: 60
        ScaleOutCooldown: 60
        PredefinedMetricSpecification:
          PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Créer une stratégie de mise à l'échelle pour un service Amazon ECS (métriques : CPU et mémoire moyens)

Cet extrait montre comment créer une stratégie et l'appliquer à une ressource [AWS::ECS::Service](#) à l'aide de la ressource [AWS::ApplicationAutoScaling::ScalingPolicy](#). La ressource [AWS::ApplicationAutoScaling::ScalableTarget](#) déclare une cible évolutive à laquelle cette politique est appliquée. Application Auto Scaling peut ajuster le nombre de tâches à un minimum de 1 tâche et un maximum de 6 tâches.

Cet exemple crée deux politiques de mise à l'échelle avec le type de politique `TargetTrackingScaling`. Les politiques sont utilisées pour mettre à l'échelle le service ECS en fonction de l'utilisation moyenne du processeur et de la mémoire du service. Il utilise les fonctions intrinsèques `Fn::Join` et `Ref` pour construire la propriété `ResourceId` avec les noms logiques des ressources [AWS::ECS::Cluster](#) (`myContainerCluster`) et `AWS::ECS::Service` (`myService`) spécifiées dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

JSON

```
{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : "6",
        "MinCapacity" : "1",
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",
        "ScalableDimension" : "ecs:service:DesiredCount",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "service",
              {
                "Ref" : "myContainerCluster"
              },
              {
                "Fn::GetAtt" : [
                  "myService",
                  "Name"
                ]
              }
            ]
          ]
        }
      }
    },
    "ServiceScalingPolicyCPU" : {
```

```

    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
      "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu70" },
      "PolicyType" : "TargetTrackingScaling",
      "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
      "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : 70.0,
        "ScaleInCooldown" : 180,
        "ScaleOutCooldown" : 60,
        "PredefinedMetricSpecification" : {
          "PredefinedMetricType" : "ECSServiceAverageCPUUtilization"
        }
      }
    },
    "ServiceScalingPolicyMem" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-mem90" },
        "PolicyType" : "TargetTrackingScaling",
        "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 90.0,
          "ScaleInCooldown" : 180,
          "ScaleOutCooldown" : 60,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "ECSServiceAverageMemoryUtilization"
          }
        }
      }
    }
  }
}

```

YAML

```

---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 6
      MinCapacity: 1

```

```

RoleARN:
  Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
ServiceNamespace: ecs
ScalableDimension: 'ecs:service:DesiredCount'
ResourceId: !Join
  - /
  - - service
    - !Ref myContainerCluster
    - !GetAtt myService.Name
ServiceScalingPolicyCPU:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu70
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ECSScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 70.0
      ScaleInCooldown: 180
      ScaleOutCooldown: 60
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageCPUUtilization
ServiceScalingPolicyMem:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub ${AWS::StackName}-target-tracking-mem90
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ECSScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 90.0
      ScaleInCooldown: 180
      ScaleOutCooldown: 60
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageMemoryUtilization

```

Créer une politique de mise à l'échelle pour un service Amazon ECS (métrique : nombre moyen de requêtes par cible)

L'exemple suivant applique une politique de mise à l'échelle de suivi cible avec la métrique prédéfinie `ALBRequestCountPerTarget` à un service ECS. La politique est utilisée pour ajouter de la capacité au service ECS lorsque le nombre de demandes par cible dépasse la valeur cible. La valeur de `DisableScaleIn` étant définie sur `true`, la politique de suivi des cibles n'entame pas la capacité de la cible évolutive.

Il utilise les fonctions intrinsèques `Fn::Join` et `Fn::GetAtt` pour construire la propriété `ResourceLabel` avec les noms logiques des ressources [AWS::ElasticLoadBalancingV2::LoadBalancer](#) (`myLoadBalancer`) et [AWS::ElasticLoadBalancingV2::TargetGroup](#) (`myTargetGroup`) spécifiées dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

Les propriétés `MaxCapacity` et `MinCapacity` de la cible évolutive et de la propriété `TargetValue` des valeurs des paramètres de référence de politique de mise à l'échelle que vous transmettez au modèle lors de la création ou de la mise à jour d'une pile.

JSON

```
{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : { "Ref" : "MaxCount" },
        "MinCapacity" : { "Ref" : "MinCount" },
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",
        "ScalableDimension" : "ecs:service:DesiredCount",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "service",
              {
                "Ref" : "myContainerCluster"
              },
              {
                "Fn::GetAtt" : [
                  "myService",
                  "Name"
                ]
              }
            ]
          ]
        }
      }
    }
  }
},
```

```

"ServiceScalingPolicyALB" : {
  "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
  "Properties" : {
    "PolicyName" : "alb-requests-per-target-per-minute",
    "PolicyType" : "TargetTrackingScaling",
    "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
    "TargetTrackingScalingPolicyConfiguration" : {
      "TargetValue" : { "Ref" : "ALBPolicyTargetValue" },
      "ScaleInCooldown" : 180,
      "ScaleOutCooldown" : 30,
      "DisableScaleIn" : true,
      "PredefinedMetricSpecification" : {
        "PredefinedMetricType" : "ALBRequestCountPerTarget",
        "ResourceLabel" : {
          "Fn::Join" : [
            "/",
            [
              {
                "Fn::GetAtt" : [
                  "myLoadBalancer",
                  "LoadBalancerFullName"
                ]
              },
              {
                "Fn::GetAtt" : [
                  "myTargetGroup",
                  "TargetGroupFullName"
                ]
              }
            ]
          ]
        }
      }
    }
  }
}

```

YAML

```

---
Resources:

```

```

ECSScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: !Ref MaxCount
    MinCapacity: !Ref MinCount
    RoleARN:
      Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
    ServiceNamespace: ecs
    ScalableDimension: 'ecs:service:DesiredCount'
    ResourceId: !Join
      - /
      - - service
        - !Ref myContainerCluster
        - !GetAtt myService.Name
ServiceScalingPolicyALB:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: alb-requests-per-target-per-minute
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ECSScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: !Ref ALBPolicyTargetValue
      ScaleInCooldown: 180
      ScaleOutCooldown: 30
      DisableScaleIn: true
      PredefinedMetricSpecification:
        PredefinedMetricType: ALBRequestCountPerTarget
        ResourceLabel: !Join
          - '/'
          - - !GetAtt myLoadBalancer.LoadBalancerFullName
            - !GetAtt myTargetGroup.TargetGroupFullName

```

Créer une action planifiée avec une expression cron pour une fonction Lambda

Cet extrait enregistre la concurrence provisionnée pour un alias de fonction ([AWS::Lambda::Alias](#)) nommé BLUE à l'aide de la ressource [AWS::ApplicationAutoScaling::ScalableTarget](#). Il crée également une action planifiée avec une planification récurrente à l'aide d'une expression cron. Le fuseau horaire du calendrier récurrent est UTC.

Il utilise les fonctions intrinsèques `Fn::Join` et `Ref` de la propriété `RoleARN` pour spécifier l'ARN du rôle lié au service. Il utilise la fonction intrinsèque `Fn::Sub` pour construire la propriété `ResourceId` avec le nom logique de la ressource [AWS::Lambda::Function](#) ou

[AWS::Serverless::Function](#) spécifiée dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

Note

Vous ne pouvez pas allouer la simultanéité provisionnée sur un alias qui pointe vers la version non publiée (\$LATEST).

Pour plus d'informations sur la création d'un CloudFormation modèle pour les ressources Lambda, consultez le billet de blog [Scheduling Lambda Provisioned Concurrency for AWS Lambda Provisioned Concurrency for Recurrent Peak Usage](#) sur le Compute Blog. AWS

JSON

```
{
  "ScalableTarget" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties" : {
      "MaxCapacity" : 250,
      "MinCapacity" : 0,
      "RoleARN" : {
        "Fn::Join" : [
          ":",
          [
            "arn:aws:iam:",
            {
              "Ref" : "AWS::AccountId"
            },
            "role/aws-service-role/lambda.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency"
          ]
        ]
      },
    },
    "ServiceNamespace" : "lambda",
    "ScalableDimension" : "lambda:function:ProvisionedConcurrency",
    "ResourceId" : { "Fn::Sub" : "function:${logicalName}:BLUE" },
    "ScheduledActions" : [
      {
        "ScalableTargetAction" : {
          "MinCapacity" : "250"
        },
        "ScheduledActionName" : "my-scale-out-scheduled-action",
      }
    ]
  }
}
```

```

        "Schedule" : "cron(0 18 * * ? *)",
        "EndTime" : "2022-12-31T12:00:00.000Z"
    }
}
}
}
}

```

YAML

```

ScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: 250
    MinCapacity: 0
    RoleARN: !Join
      - ':'
      - - 'arn:aws:iam:'
        - !Ref 'AWS::AccountId'
        - role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
  AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency
    ServiceNamespace: lambda
    ScalableDimension: lambda:function:ProvisionedConcurrency
    ResourceId: !Sub function:${logicalName}:BLUE
    ScheduledActions:
      - ScalableTargetAction:
          MinCapacity: 250
          ScheduledActionName: my-scale-out-scheduled-action
          Schedule: 'cron(0 18 * * ? *)'
          EndTime: '2022-12-31T12:00:00.000Z'

```

Créer une action planifiée avec une expression **at** pour un parc d'instances Spot

Cet extrait montre comment créer deux actions planifiées qui ne se produisent qu'une seule fois pour une ressource [AWS::EC2::SpotFleet](#) utilisant la ressource [AWS::ApplicationAutoScaling::ScalableTarget](#). Le fuseau horaire de chaque action programmée ponctuelle est UTC.

Il utilise les fonctions intrinsèques `Fn::Join` et `Ref` pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::EC2::SpotFleet` spécifiée dans le même modèle. Pour plus d'informations, consultez la [Référence des fonctions intrinsèques](#).

Note

La demande de parc d'instances Spot doit être de type `maintain`. Le dimensionnement automatique n'est pas pris en charge pour les demandes uniques et les blocs d'instances Spot.

JSON

```
{
  "Resources" : {
    "SpotFleetScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 0,
        "MinCapacity" : 0,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest" },
        "ServiceNamespace" : "ec2",
        "ScalableDimension" : "ec2:spot-fleet-request:TargetCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "spot-fleet-request",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      },
    },
    "ScheduledActions" : [
      {
        "ScalableTargetAction" : {
          "MaxCapacity" : "10",
          "MinCapacity" : "10"
        },
        "ScheduledActionName" : "my-scale-out-scheduled-action",
        "Schedule" : "at(2022-05-20T13:00:00)"
      },
    ]
  }
}
```

```

        "ScalableTargetAction" : {
            "MaxCapacity" : "0",
            "MinCapacity" : "0"
        },
        "ScheduledActionName" : "my-scale-in-scheduled-action",
        "Schedule" : "at(2022-05-20T21:00:00)"
    }
}
}
}
}
}
}
}

```

YAML

```

---
Resources:
  SpotFleetScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 0
      MinCapacity: 0
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ec2.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest'
      ServiceNamespace: ec2
      ScalableDimension: 'ec2:spot-fleet-request:TargetCapacity'
      ResourceId: !Join
        - /
        - - spot-fleet-request
          - !Ref logicalName
      ScheduledActions:
        - ScalableTargetAction:
            MaxCapacity: 10
            MinCapacity: 10
            ScheduledActionName: my-scale-out-scheduled-action
            Schedule: 'at(2022-05-20T13:00:00)'
        - ScalableTargetAction:
            MaxCapacity: 0
            MinCapacity: 0
            ScheduledActionName: my-scale-in-scheduled-action
            Schedule: 'at(2022-05-20T21:00:00)'

```

AWS Extraits du modèle de console de facturation

Cet exemple crée un plan tarifaire avec une règle de tarification de marge globale de 10 %. Ce plan tarifaire est rattaché au groupe de facturation. Le groupe de facturation comporte également deux postes de facturation personnalisés qui appliquent des frais de 10 USD et des frais de 10 % en plus du coût total du groupe de facturation.

JSON

```
{
  "Parameters": {
    "LinkedAccountIds": {
      "Type": "ListNumber"
    },
    "PrimaryAccountId": {
      "Type": "Number"
    }
  },
  "Resources": {
    "TestPricingRule": {
      "Type": "AWS::BillingConductor::PricingRule",
      "Properties": {
        "Name": "TestPricingRule",
        "Description": "Test pricing rule created through Cloudformation. Mark everything by 10%.",
        "Type": "MARKUP",
        "Scope": "GLOBAL",
        "ModifierPercentage": 10
      }
    },
    "TestPricingPlan": {
      "Type": "AWS::BillingConductor::PricingPlan",
      "Properties": {
        "Name": "TestPricingPlan",
        "Description": "Test pricing plan created through Cloudformation.",
        "PricingRuleArns": [
          {"Fn::GetAtt": ["TestPricingRule", "Arn"]}
        ]
      }
    },
    "TestBillingGroup": {
      "Type": "AWS::BillingConductor::BillingGroup",
      "Properties": {
```



```

    "Name": "TestBillingGroup",
    "Description": "Test billing group created through Cloudformation with 1
linked account. The linked account is also the primary account.",
    "PrimaryAccountId": {
        "Ref": "PrimaryAccountId"
    },
    "AccountGrouping": {
        "LinkedAccountIds": null
    },
    "ComputationPreference": {
        "PricingPlanArn": {
            "Fn::GetAtt": ["TestPricingPlan", "Arn"]
        }
    }
},
"TestFlatCustomLineItem": {
    "Type": "AWS::BillingConductor::CustomLineItem",
    "Properties": {
        "Name": "TestFlatCustomLineItem",
        "Description": "Test flat custom line item created through Cloudformation
for a $10 charge.",
        "BillingGroupArn": {
            "Fn::GetAtt": ["TestBillingGroup", "Arn"]
        },
        "CustomLineItemChargeDetails": {
            "Flat": {
                "ChargeValue": 10
            },
            "Type": "FEE"
        }
    }
},
"TestPercentageCustomLineItem": {
    "Type": "AWS::BillingConductor::CustomLineItem",
    "Properties": {
        "Name": "TestPercentageCustomLineItem",
        "Description": "Test percentage custom line item created through
Cloudformation for a %10 additional charge on the overall total bill of the billing
group.",
        "BillingGroupArn": {
            "Fn::GetAtt": ["TestBillingGroup", "Arn"]
        },
        "CustomLineItemChargeDetails": {

```



```
PrimaryAccountId: !Ref PrimaryAccountId
AccountGrouping:
  LinkedAccountIds: !Ref LinkedAccountIds
ComputationPreference:
  PricingPlanArn: !GetAtt TestPricingPlan.Arn
TestFlatCustomLineItem:
  Type: AWS::BillingConductor::CustomLineItem
  Properties:
    Name: 'TestFlatCustomLineItem'
    Description: 'Test flat custom line item created through Cloudformation for a $10
charge.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn
    CustomLineItemChargeDetails:
      Flat:
        ChargeValue: 10
        Type: 'FEE'
TestPercentageCustomLineItem:
  Type: AWS::BillingConductor::CustomLineItem
  Properties:
    Name: 'TestPercentageCustomLineItem'
    Description: 'Test percentage custom line item created through Cloudformation for
a %10 additional charge on the overall total bill of the billing group.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn
    CustomLineItemChargeDetails:
      Percentage:
        PercentageValue: 10
        ChildAssociatedResources:
          - !GetAtt TestBillingGroup.Arn
        Type: 'FEE'
```

CloudFormation extraits de modèles

Rubriques

- [Piles imbriquées](#)
- [Condition d'attente](#)

Piles imbriquées

Imbrication d'une pile dans un modèle

Cet exemple de modèle contient une ressource de pile imbriquée appelée `myStack`. Lorsqu'il CloudFormation crée une pile à partir du modèle, il crée `lemyStack`, dont le modèle est spécifié dans la `TemplateURL` propriété. La valeur de sortie `StackRef` renvoie l'ID de pile pour `myStack` et la valeur `OutputFromNestedStack` renvoie la valeur de sortie `BucketName` à partir de la ressource `myStack`. Le format `Outputs.nestedstackoutputname` est réservé pour spécifier les valeurs de sortie des piles imbriquées et peut être utilisé n'importe où dans le modèle.

Pour de plus amples informations, veuillez consulter [AWS::CloudFormation::Stack](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template",
        "TimeoutInMinutes" : "60"
      }
    }
  },
  "Outputs": {
    "StackRef": {"Value": { "Ref" : "myStack"}},
    "OutputFromNestedStack" : {
      "Value" : { "Fn::GetAtt" : [ "myStack", "Outputs.BucketName" ] }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStack:
    Type: AWS::CloudFormation::Stack
    Properties:
```

```
TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template
TimeoutInMinutes: '60'
Outputs:
  StackRef:
    Value: !Ref myStack
  OutputFromNestedStack:
    Value: !GetAtt myStack.Outputs.BucketName
```

Imbrication d'une pile avec des paramètres d'entrée dans un modèle

Cet exemple de modèle contient une ressource de pile qui spécifie les paramètres d'entrée. Lors de la CloudFormation création d'une pile à partir de ce modèle, il utilise les paires de valeurs déclarées dans la Parameters propriété comme paramètres d'entrée pour le modèle utilisé pour créer la myStackWithParams pile. Dans cet exemple, les paramètres InstanceType et KeyName sont spécifiés.

Pour de plus amples informations, veuillez consulter [AWS::CloudFormation::Stack](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStackWithParams" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-
east-1/EC2ChooseAMI.template",
        "Parameters" : {
          "InstanceType" : "t2.micro",
          "KeyName" : "mykey"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStackWithParams:
```

```
Type: AWS::CloudFormation::Stack
Properties:
  TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
  EC2ChooseAMI.template
  Parameters:
    InstanceType: t2.micro
    KeyName: mykey
```

Condition d'attente

Utilisation d'une condition d'attente avec une instance Amazon EC2

Important

Pour les ressources Amazon EC2 et Auto Scaling, nous vous recommandons d'utiliser un `CreationPolicy` attribut plutôt que des conditions d'attente. Ajoutez un `CreationPolicy` attribut à ces ressources et utilisez le script d'assistance `cfn-signal` pour signaler qu'un processus de création d'instance s'est terminé avec succès.

Si vous ne pouvez pas utiliser une politique de création, affichez l'exemple de modèle suivant, qui déclare une instance Amazon EC2 avec une condition d'attente. La condition d'attente `myWaitCondition` utilise `myWaitConditionHandle` pour la signalisation, utilise l'attribut `DependsOn` pour spécifier que la condition d'attente se déclenchera après la création de la ressource d'instance Amazon EC2, et utilise la propriété `Timeout` pour spécifier une durée de 4 500 secondes pour la condition d'attente. De plus, l'URL pré-signée qui signale la condition d'attente est transmise à l'instance Amazon EC2 avec la propriété `UserData` de la ressource `Ec2Instance`, permettant ainsi à une application ou à un script s'exécutant sur cette instance Amazon EC2 de récupérer l'URL pré-signée et de l'utiliser pour signaler la réussite ou l'échec de la condition d'attente. Vous devez utiliser `cfn-signal` ou créer l'application ou le script qui signale la condition d'attente. La valeur de sortie `ApplicationData` contient les données renvoyées par le signal de condition d'attente.

Pour de plus amples informations, veuillez consulter [Création de conditions d'attente dans un CloudFormation modèle](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
```

```

    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0123456789abcdef0"
      },
      "us-west-1" : {
        "AMI" : "ami-0987654321fedcba0"
      },
      "eu-west-1" : {
        "AMI" : "ami-0abcdef123456789a"
      },
      "ap-northeast-1" : {
        "AMI" : "ami-0fedcba987654321b"
      },
      "ap-southeast-1" : {
        "AMI" : "ami-0c1d2e3f4a5b6c7d8"
      }
    },
    "Resources" : {
      "Ec2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
          "UserData" : { "Fn::Base64" : { "Ref" : "myWaitHandle" } },
          "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ] }
        }
      },
      "myWaitHandle" : {
        "Type" : "AWS::CloudFormation::WaitConditionHandle",
        "Properties" : {
        }
      },
      "myWaitCondition" : {
        "Type" : "AWS::CloudFormation::WaitCondition",
        "DependsOn" : "Ec2Instance",
        "Properties" : {
          "Handle" : { "Ref" : "myWaitHandle" },
          "Timeout" : "4500"
        }
      }
    },
    "Outputs" : {
      "ApplicationData" : {
        "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ] },

```

```
        "Description" : "The data passed back as part of signalling the
WaitCondition."
    }
}
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0123456789abcdef0
    us-west-1:
      AMI: ami-0987654321fedcba0
    eu-west-1:
      AMI: ami-0abcdef123456789a
    ap-northeast-1:
      AMI: ami-0fedcba987654321b
    ap-southeast-1:
      AMI: ami-0c1d2e3f4a5b6c7d8
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
  myWaitHandle:
    Type: AWS::CloudFormation::WaitConditionHandle
    Properties: {}
  myWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    DependsOn: Ec2Instance
    Properties:
      Handle: !Ref myWaitHandle
      Timeout: '4500'
Outputs:
  ApplicationData:
```


Value: !GetAtt myWaitCondition.Data

Description: The data passed back as part of signalling the WaitCondition.

Utilisation du script d'assistant cfn-signal pour envoyer des signaux à une condition d'attente

Cet exemple montre une ligne de commande `cfn-signal` qui signale la réussite d'une condition d'attente. Vous devez définir la ligne de commande dans la propriété `UserData` de l'instance EC2.

JSON

```
"UserData": {
  "Fn::Base64": {
    "Fn::Join": [
      "",
      [
        "#!/bin/bash -xe\n",
        "/opt/aws/bin/cfn-signal --exit-code 0 '",
        {
          "Ref": "myWaitHandle"
        },
        "'\n"
      ]
    ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    /opt/aws/bin/cfn-signal --exit-code 0 '${myWaitHandle}'
```

Utilisation de Curl pour envoyer des signaux à une condition d'attente

Cet exemple présente une ligne de commande `Curl` qui envoie un signal de réussite à une condition d'attente.

```
curl -T /tmp/a "https://cloudformation-waitcondition-test.s3.amazonaws.com/
arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW"
```

```
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo%3D"
```

Où le fichier/tmp/a contient la structure JSON suivante :

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

Cet exemple montre une ligne de commande Curl qui envoie le même signal de réussite, sauf qu'il envoie le code JSON comme paramètre dans la ligne de commande.

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" : "Configuration Complete","UniqueId" : "ID1234","Data" : "Application has completed configuration."}' "https://cloudformation-waitcondition-test.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo%3D"
```

Extraits CloudFront de modèles Amazon

Utilisez ces exemples d'extraits de modèle avec votre ressource de CloudFront distribution Amazon dans. CloudFormation Pour plus d'informations, consultez la [référence CloudFront des types de ressources Amazon](#).

Rubriques

- [Ressource CloudFront de distribution Amazon avec une origine Amazon S3](#)
- [Ressource CloudFront de distribution Amazon avec origine personnalisée](#)
- [CloudFront Distribution Amazon avec prise en charge de plusieurs origines](#)
- [CloudFront Distribution Amazon avec une fonction Lambda comme origine](#)
- [Consultez aussi](#)

Ressource CloudFront de distribution Amazon avec une origine Amazon S3

L'exemple de modèle suivant montre une CloudFront [distribution](#) Amazon utilisant un [S3Origin](#) et une ancienne identité d'accès à l'origine (OAI). Pour plus d'informations sur l'utilisation du contrôle d'accès à l'origine (OAC) à la place, consultez [Restreindre l'accès à une origine Amazon Simple Storage Service](#) dans le manuel Amazon CloudFront Developer Guide.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "Id" : "myS3Origin",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          } ],
          "Enabled" : "true",
          "Comment" : "Some comment",
          "DefaultRootObject" : "index.html",
          "Logging" : {
            "IncludeCookies" : "false",
            "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
            "Prefix" : "myprefix"
          },
          "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
          "DefaultCacheBehavior" : {
            "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
            "TargetOriginId" : "myS3Origin",
            "ForwardedValues" : {
              "QueryString" : "false",
              "Cookies" : { "Forward" : "none" }
            },
            "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
            "ViewerProtocolPolicy" : "allow-all"
          }
        }
      }
    }
  }
}
```

```
    },  
    "PriceClass" : "PriceClass_200",  
    "Restrictions" : {  
        "GeoRestriction" : {  
            "RestrictionType" : "whitelist",  
            "Locations" : [ "AQ", "CV" ]  
        }  
    },  
    "ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }  
}  
  
}  
  
}
```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'

Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
            Id: myS3Origin
            S3OriginConfig:
              OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
        Enabled: 'true'
        Comment: Some comment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'false'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
          - mysite.example.com
          - yoursite.example.com
        DefaultCacheBehavior:
          AllowedMethods:
            - DELETE
            - GET
            - HEAD

```

```

- OPTIONS
- PATCH
- POST
- PUT
TargetOriginId: myS3Origin
ForwardedValues:
  QueryString: 'false'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
- AQ
- CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'

```

Ressource CloudFront de distribution Amazon avec origine personnalisée

L'exemple de modèle suivant montre une CloudFront [distribution](#) Amazon utilisant un [CustomOrigin](#).

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "www.example.com",
            "Id" : "myCustomOrigin",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          }
        ]
      }
    }
  }
}

```

```

    } ],
    "Enabled" : "true",
    "Comment" : "Somecomment",
    "DefaultRootObject" : "index.html",
    "Logging" : {
        "IncludeCookies" : "true",
        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
        "Prefix" : "myprefix"
    },
    "Aliases" : [
        "mysite.example.com",
        "*.yoursite.example.com"
    ],
    "DefaultCacheBehavior" : {
        "TargetOriginId" : "myCustomOrigin",
        "SmoothStreaming" : "false",
        "ForwardedValues" : {
            "QueryString" : "false",
            "Cookies" : { "Forward" : "all" }
        },
        "TrustedSigners" : [
            "1234567890EX",
            "1234567891EX"
        ],
        "ViewerProtocolPolicy" : "allow-all"
    },
    "CustomErrorResponses" : [ {
        "ErrorCode" : "404",
        "ResponsePagePath" : "/error-pages/404.html",
        "ResponseCode" : "200",
        "ErrorCachingMinTTL" : "30"
    } ],
    "PriceClass" : "PriceClass_200",
    "Restrictions" : {
        "GeoRestriction" : {
            "RestrictionType" : "whitelist",
            "Locations" : [ "AQ", "CV" ]
        }
    },
    "ViewerCertificate": { "CloudFrontDefaultCertificate" : "true" }
}
}
}
}
}

```

```
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: www.example.com
            Id: myCustomOrigin
            CustomOriginConfig:
              HTTPPort: '80'
              HTTPSPort: '443'
              OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Somecomment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
          - mysite.example.com
          - "*.yoursite.example.com"
        DefaultCacheBehavior:
          TargetOriginId: myCustomOrigin
          SmoothStreaming: 'false'
          ForwardedValues:
            QueryString: 'false'
            Cookies:
              Forward: all
          TrustedSigners:
            - 1234567890EX
            - 1234567891EX
          ViewerProtocolPolicy: allow-all
      CustomErrorResponses:
        - ErrorCode: '404'
          ResponsePagePath: "/error-pages/404.html"
          ResponseCode: '200'
          ErrorCachingMinTTL: '30'
```

```

PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
      - AQ
      - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'

```

CloudFront Distribution Amazon avec prise en charge de plusieurs origines

L'exemple de modèle suivant montre comment déclarer une CloudFront [distribution prenant](#) en charge plusieurs origines. Dans le [DistributionConfig](#), une liste d'origines est fournie et un [DefaultCacheBehavior](#) est défini.

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "Id" : "myS3Origin",
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          },
          {
            "Id" : "myCustomOrigin",
            "DomainName" : "www.example.com",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          }
        ]
      }
    }
  }
}

```



```

    "Enabled" : "true",
    "Comment" : "Some comment",
    "DefaultRootObject" : "index.html",
    "Logging" : {
        "IncludeCookies" : "true",
        "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
        "Prefix" : "myprefix"
    },
    "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
    "DefaultCacheBehavior" : {
        "TargetOriginId" : "myS3Origin",
        "ForwardedValues" : {
            "QueryString" : "false",
            "Cookies" : { "Forward" : "all" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "100",
        "SmoothStreaming" : "true"
    },
    "CacheBehaviors" : [ {
        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myS3Origin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",
        "PathPattern" : "images1/*.jpg"
    },
    {
        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myCustomOrigin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",

```

```
        "PathPattern" : "images2/*.jpg"
    },
],
"CustomErrorResponses" : [ {
    "ErrorCode" : "404",
    "ResponsePagePath" : "/error-pages/404.html",
    "ResponseCode" : "200",
    "ErrorCachingMinTTL" : "30"
} ],
"PriceClass" : "PriceClass_All",
"ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
}
}
```

YAML

```

AWS::CloudFront::Distribution
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - Id: myS3Origin
            DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
            S3OriginConfig:
              OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
          - Id: myCustomOrigin
            DomainName: www.example.com
            CustomOriginConfig:
              HTTPPort: '80'
              HTTPSPort: '443'
              OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Some comment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix

```

```
Aliases:
- mysite.example.com
- yoursite.example.com
DefaultCacheBehavior:
  TargetOriginId: myS3Origin
  ForwardedValues:
    QueryString: 'false'
    Cookies:
      Forward: all
  TrustedSigners:
    - 1234567890EX
    - 1234567891EX
  ViewerProtocolPolicy: allow-all
  MinTTL: '100'
  SmoothStreaming: 'true'
CacheBehaviors:
- AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myS3Origin
  ForwardedValues:
    QueryString: 'true'
    Cookies:
      Forward: none
  TrustedSigners:
    - 1234567890EX
    - 1234567891EX
  ViewerProtocolPolicy: allow-all
  MinTTL: '50'
  PathPattern: images1/*.jpg
- AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myCustomOrigin
```

```
ForwardedValues:
  QueryString: 'true'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '50'
PathPattern: images2/*.jpg
CustomErrorResponses:
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_All
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

CloudFront Distribution Amazon avec une fonction Lambda comme origine

L'exemple suivant crée une CloudFront distribution qui fait face à une URL de fonction Lambda spécifiée (fournie en tant que paramètre), permettant l'accès, la mise en cache, la compression et la diffusion globale uniquement via HTTPS. Il configure l'URL Lambda en tant qu'origine HTTPS personnalisée et applique une politique de mise en cache AWS standard. La distribution est optimisée pour les performances avec HTTP/2 et IPv6 prend en charge et génère le nom de CloudFront domaine, permettant aux utilisateurs d'accéder à la fonction Lambda via un point de terminaison sécurisé soutenu par un CDN. Pour plus d'informations, consultez la section [Utiliser Amazon CloudFront AWS Lambda comme origine pour accélérer vos applications Web](#) sur le AWS blog.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "LambdaEndpoint": {
      "Type": "String",
      "Description": "The Lambda function URL endpoint without the 'https://'"
    }
  },
  "Resources": {
```

```

    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "PriceClass": "PriceClass_All",
          "HttpVersion": "http2",
          "IPV6Enabled": true,
          "Origins": [
            {
              "DomainName": {
                "Ref": "LambdaEndpoint"
              },
              "Id": "LambdaOrigin",
              "CustomOriginConfig": {
                "HTTPSPort": 443,
                "OriginProtocolPolicy": "https-only"
              }
            }
          ],
          "Enabled": "true",
          "DefaultCacheBehavior": {
            "TargetOriginId": "LambdaOrigin",
            "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
            "ViewerProtocolPolicy": "redirect-to-https",
            "SmoothStreaming": "false",
            "Compress": "true"
          }
        }
      }
    },
    "Outputs": {
      "CloudFrontDomain": {
        "Description": "CloudFront default domain name configured",
        "Value": {
          "Fn::Sub": "https://${MyDistribution.DomainName}/"
        }
      }
    }
  }
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  LambdaEndpoint:
    Type: String
    Description: The Lambda function URL endpoint without the 'https://'
Resources:
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        PriceClass: PriceClass_All
        HttpVersion: http2
        IPV6Enabled: true
        Origins:
          - DomainName: !Ref LambdaEndpoint
            Id: LambdaOrigin
            CustomOriginConfig:
              HTTPSPort: 443
              OriginProtocolPolicy: https-only
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: LambdaOrigin
          CachePolicyId: '658327ea-f89d-4fab-a63d-7e88639e58f6'
          ViewerProtocolPolicy: redirect-to-https
          SmoothStreaming: 'false'
          Compress: 'true'
Outputs:
  CloudFrontDomain:
    Description: CloudFront default domain name configured
    Value: !Sub https://${MyDistribution.DomainName}/
```

Consultez aussi

Pour un exemple d'ajout d'un alias personnalisé à un enregistrement Route 53 afin de donner un nom convivial à une CloudFront distribution, consultez [Enregistrement de ressource Alias défini pour une CloudFront distribution](#).

Extraits CloudWatch de modèles Amazon

Utilisez ces exemples d'extraits de modèles pour décrire vos CloudWatch ressources Amazon dans CloudFormation. Pour plus d'informations, consultez la [référence CloudWatch des types de ressources Amazon](#).

Rubriques

- [Alarme de facturation](#)
- [Alarme d'utilisation de l'UC](#)
- [Récupération d'une instance Amazon Elastic Compute Cloud](#)
- [Création d'un tableau de bord de base](#)
- [Création d'un tableau de bord avec des side-by-side widgets](#)

Alarme de facturation

Dans l'exemple suivant, Amazon CloudWatch envoie une notification par e-mail lorsque les frais prélevés sur votre AWS compte dépassent le seuil d'alarme. Pour recevoir des notifications d'utilisation, activez les alertes de facturation. Pour plus d'informations, consultez la section [Créer une alarme de facturation pour surveiller vos AWS frais estimés](#) dans le guide de CloudWatch l'utilisateur Amazon. >

JSON

```
"SpendingAlarm": {
  "Type": "AWS::CloudWatch::Alarm",
  "Properties": {
    "AlarmDescription": { "Fn::Join": ["", [
      "Alarm if AWS spending is over $",
      { "Ref": "AlarmThreshold" }
    ] ] },
    "Namespace": "AWS/Billing",
    "MetricName": "EstimatedCharges",
    "Dimensions": [{
      "Name": "Currency",
      "Value": "USD"
    }],
    "Statistic": "Maximum",
    "Period": "21600",
    "EvaluationPeriods": "1",
```

```

    "Threshold": { "Ref": "AlarmThreshold" },
    "ComparisonOperator": "GreaterThanThreshold",
    "AlarmActions": [{
        "Ref": "BillingAlarmNotification"
    }],
    "InsufficientDataActions": [{
        "Ref": "BillingAlarmNotification"
    }]
  }
}

```

YAML

```

SpendingAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription:
      'Fn::Join':
        - ''
        - - Alarm if AWS spending is over $
          - !Ref: AlarmThreshold
    Namespace: AWS/Billing
    MetricName: EstimatedCharges
    Dimensions:
      - Name: Currency
        Value: USD
    Statistic: Maximum
    Period: '21600'
    EvaluationPeriods: '1'
    Threshold:
      !Ref: "AlarmThreshold"
    ComparisonOperator: GreaterThanThreshold
    AlarmActions:
      - !Ref: "BillingAlarmNotification"
    InsufficientDataActions:
      - !Ref: "BillingAlarmNotification"

```

Alarme d'utilisation de l'UC

L'exemple d'extrait suivant crée une alarme qui envoie une notification quand l'utilisation d'UC moyenne d'une instance Amazon EC2 dépasse 90 % pendant plus de 60 secondes pendant trois périodes d'évaluation.

JSON

```
"CPUAlarm" : {
  "Type" : "AWS::CloudWatch::Alarm",
  "Properties" : {
    "AlarmDescription" : "CPU alarm for my instance",
    "AlarmActions" : [ { "Ref" : "logical name of an AWS::SNS::Topic resource" } ],
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "60",
    "EvaluationPeriods" : "3",
    "Threshold" : "90",
    "ComparisonOperator" : "GreaterThanOrEqualToThreshold",
    "Dimensions" : [ {
      "Name" : "InstanceId",
      "Value" : { "Ref" : "logical name of an AWS::EC2::Instance resource" }
    } ]
  }
}
```

YAML

```
CPUAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: CPU alarm for my instance
    AlarmActions:
      - !Ref: "logical name of an AWS::SNS::Topic resource"
    MetricName: CPUUtilization
    Namespace: AWS/EC2
    Statistic: Average
    Period: '60'
    EvaluationPeriods: '3'
    Threshold: '90'
    ComparisonOperator: GreaterThanOrEqualToThreshold
    Dimensions:
      - Name: InstanceId
        Value: !Ref: "logical name of an AWS::EC2::Instance resource"
```

Récupération d'une instance Amazon Elastic Compute Cloud

L' CloudWatch alarme suivante permet de récupérer une instance EC2 lorsque le contrôle de statut échoue pendant 15 minutes consécutives. Pour plus d'informations sur les actions d'alarme, consultez la section [Créer des alarmes pour arrêter, arrêter, redémarrer ou récupérer une instance EC2](#) dans le guide de l' CloudWatch utilisateur Amazon.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "RecoveryInstance" : {
      "Description" : "The EC2 instance ID to associate this alarm with.",
      "Type" : "AWS::EC2::Instance::Id"
    }
  },
  "Resources": {
    "RecoveryTestAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Trigger a recovery when instance status check fails for 15 consecutive minutes.",
        "Namespace": "AWS/EC2" ,
        "MetricName": "StatusCheckFailed_System",
        "Statistic": "Minimum",
        "Period": "60",
        "EvaluationPeriods": "15",
        "ComparisonOperator": "GreaterThanThreshold",
        "Threshold": "0",
        "AlarmActions": [ {"Fn::Join" : [ "", [ "arn:aws:automate:", { "Ref" : "AWS::Region" }, ":ec2:recover" ] ] } ],
        "Dimensions": [{"Name": "InstanceId", "Value": {"Ref": "RecoveryInstance"}}]
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
```

```

RecoveryInstance:
  Description: The EC2 instance ID to associate this alarm with.
  Type: AWS::EC2::Instance::Id
Resources:
  RecoveryTestAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: Trigger a recovery when instance status check fails for 15
        consecutive minutes.
      Namespace: AWS/EC2
      MetricName: StatusCheckFailed_System
      Statistic: Minimum
      Period: '60'
      EvaluationPeriods: '15'
      ComparisonOperator: GreaterThanThreshold
      Threshold: '0'
      AlarmActions: [ !Sub "arn:aws:automate:${AWS::Region}:ec2:recover" ]
      Dimensions:
        - Name: InstanceId
          Value: !Ref: RecoveryInstance

```

Création d'un tableau de bord de base

L'exemple suivant crée un CloudWatch tableau de bord simple avec un widget métrique affichant l'utilisation du processeur et un widget texte affichant un message.

JSON

```

{
  "BasicDashboard": {
    "Type": "AWS::CloudWatch::Dashboard",
    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\n  \"widgets\": [\n    {\n      \"type\": \"metric\",\n      \"x\": 0,\n      \"y\": 0,\n      \"width\": 12,\n      \"height\": 6,\n      \"properties\": {\n        \"metrics\": [\n          [\n            \"AWS/EC2\",\n            \"CPUUtilization\",\n            \"InstanceId\",\n            \"i-012345\"\n          ],\n          {\n            \"period\": 300,\n            \"stat\": \"Average\",\n            \"region\": \"us-east-1\",\n            \"title\": \"EC2 Instance CPU\"\n          }\n        ]\n      },\n      {\n        \"type\": \"text\",\n        \"x\": 0,\n        \"y\": 7,\n        \"width\": 3,\n        \"height\": 3,\n        \"properties\": {\n          \"markdown\": \"Hello world\"\n        }\n      }\n    ]\n  }\n}"
    }
  }
}

```

YAML

```
BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]], "period":300, "stat":"Average", "region":"us-
east-1", "title":"EC2 Instance CPU"}},
{"type":"text","x":0,"y":7,"width":3,"height":3,"properties":{"markdown":"Hello
world"}}]}'
```

Création d'un tableau de bord avec des side-by-side widgets

L'exemple suivant permet de créer un tableau de bord avec deux widgets de métrique s'affichant côte à côte.

JSON

```
{
  "DashboardSideBySide": {
    "Type": "AWS::CloudWatch::Dashboard",
    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\"widgets\": [{\"type\": \"metric\", \"x\": 0, \"y\": 0,
\\\"width\\\": 12, \\\"height\\\": 6, \\\"properties\\\": {\\\"metrics\\\": [\\\"AWS/EC2\\\", \\\"CPUUtilization
\\\", \\\"InstanceId\\\", \\\"i-012345\\\"]], \\\"period\\\": 300, \\\"stat\\\": \\\"Average\\\", \\\"region\\\":
\\\"us-east-1\\\", \\\"title\\\": \\\"EC2 Instance CPU\\\"}}, {\"type\": \"metric\", \"x\": 12, \"y\": 0,
\\\"width\\\": 12, \\\"height\\\": 6, \\\"properties\\\": {\\\"metrics\\\": [\\\"AWS/S3\\\", \\\"BucketSizeBytes\\\",
\\\"BucketName\\\", \\\"amzn-s3-demo-bucket\\\"]], \\\"period\\\": 86400, \\\"stat\\\": \\\"Maximum\\\", \\\"region
\\\": \\\"us-east-1\\\", \\\"title\\\": \\\"amzn-s3-demo-bucket bytes\\\"}}]}"
    }
  }
}
```

YAML

```
DashboardSideBySide:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
```

```
DashboardBody: '{"widgets":  
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/  
EC2","CPUUtilization","InstanceId","i-012345"]], "period":300, "stat":"Average", "region":"us-  
east-1", "title":"EC2 Instance CPU"}},  
{"type":"metric","x":12,"y":0,"width":12,"height":6,"properties":  
{"metrics":[["AWS/S3","BucketSizeBytes","BucketName","amzn-s3-demo-  
bucket"]], "period":86400, "stat":"Maximum", "region":"us-east-1", "title":"amzn-s3-demo-  
bucket bytes"}}}]'
```

Extraits du modèle Amazon CloudWatch Logs

Amazon CloudWatch Logs peut surveiller votre système, votre application et vos fichiers journaux personnalisés à partir d'instances Amazon EC2 ou d'autres sources. Vous pouvez l'utiliser CloudFormation pour configurer et gérer des groupes de journaux et des filtres métriques. Pour plus d'informations sur Amazon CloudWatch Logs, consultez le [guide de l'utilisateur Amazon CloudWatch Logs](#).

Rubriques

- [Envoyer des journaux à CloudWatch Logs depuis une instance Linux](#)
- [Envoyer des journaux vers CloudWatch Logs à partir d'une instance Windows](#)
- [Consultez aussi](#)

Envoyer des journaux à CloudWatch Logs depuis une instance Linux

Le modèle suivant explique comment configurer un serveur Web sur Amazon Linux 2023 avec l'intégration CloudWatch des journaux. Le modèle effectue les tâches suivantes :

- Installe Apache et PHP.
- Configure l' CloudWatch agent pour transférer les journaux d'accès Apache vers CloudWatch Logs.
- Configure un rôle IAM pour permettre à l' CloudWatch agent d'envoyer des données de journal à CloudWatch Logs.
- Crée des alarmes et des notifications personnalisées pour surveiller les erreurs 404 ou l'utilisation élevée de la bande passante.

Les événements du journal provenant du serveur Web fournissent des données métriques pour les CloudWatch alarmes. Les deux filtres métriques décrivent comment les informations du journal sont transformées en CloudWatch métriques. La métrique 404 compte le nombre d'erreurs 404. La métrique de taille analyse la taille d'une requête. Les deux CloudWatch alarmes enverront des notifications s'il y a plus de deux 404 dans les 2 minutes ou si la taille moyenne des demandes est supérieure à 3 500 Ko en 10 minutes.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on Amazon Linux 2023 instance.",
  "Parameters": {
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "OperatorEmail": {
      "Description": "Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)",
      "Type": "String"
    }
  },
  "Resources": {
    "LogRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
```

```

        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "ec2.amazonaws.com"
                    ]
                },
                "Action": [
                    "sts:AssumeRole"
                ]
            }
        ],
        "Path": "/",
        "Policies": [
            {
                "PolicyName": "LogRolePolicy",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "logs:PutLogEvents",
                                "logs:DescribeLogStreams",
                                "logs:DescribeLogGroups",
                                "logs:CreateLogGroup",
                                "logs:CreateLogStream"
                            ],
                            "Resource": "*"
                        }
                    ]
                }
            }
        ],
        "LogRoleInstanceProfile": {
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
                "Roles": [{"Ref": "LogRole"}]
            }
        }
    },
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        }
    }
}

```

```

    }
  },
  "WebServerSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable HTTP access via port 80 and SSH access via
port 22",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "CidrIp": {"Ref": "SSHLocation"}
        }
      ]
    }
  },
  "WebServerHost": {
    "Type": "AWS::EC2::Instance",
    "Metadata": {
      "Comment": "Install a simple PHP application on Amazon Linux 2023",
      "AWS::CloudFormation::Init": {
        "config": {
          "packages": {
            "dnf": {
              "httpd": [],
              "php": [],
              "php-fpm": []
            }
          },
          "files": {
            "/etc/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.json": {
              "content": {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [{

```



```

        "file_path": "/var/log/httpd/
access_log",
        "log_group_name": {"Ref":
        "WebServerLogGroup"},
        "log_stream_name": "{instance_id}/
apache.log",
        "timestamp_format": "%d/%b/%Y:%H:
%M:%S %z"
    }
}
}
},
{
    "mode": "000644",
    "owner": "root",
    "group": "root"
},
"/var/www/html/index.php": {
    "content": "<?php\\necho '<h1>AWS CloudFormation sample
PHP application on Amazon Linux 2023</h1>';\\n?>\\n",
    "mode": "000644",
    "owner": "apache",
    "group": "apache"
},
"/etc/cfn/cfn-hup.conf": {
    "content": {
        "Fn::Join": [
            "",
            [
                "[main]\\n",
                "stack=",
                {"Ref": "AWS::StackId"},
                "\\n",
                "region=",
                {"Ref": "AWS::Region"},
                "\\n"
            ]
        ]
    },
    "mode": "000400",
    "owner": "root",
    "group": "root"
},
"/etc/cfn/hooks.d/cfn-auto-reloader.conf": {

```

```

        "content": {
            "Fn::Join": [
                "",
                [
                    "[cfn-auto-reloader-hook]\n",
                    "triggers=post.update\n",
                    "path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
                    "action=/opt/aws/bin/cfn-init -s ",
                    {"Ref": "AWS::StackId"},
                    " -r WebServerHost ",
                    " --region ",
                    {"Ref": "AWS::Region"},
                    "\n",
                    "runas=root\n"
                ]
            ]
        },
        "services": {
            "systemd": {
                "httpd": {
                    "enabled": "true",
                    "ensureRunning": "true"
                },
                "php-fpm": {
                    "enabled": "true",
                    "ensureRunning": "true"
                }
            }
        },
        "CreationPolicy": {
            "ResourceSignal": {
                "Timeout": "PT5M"
            }
        },
        "Properties": {
            "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
            "KeyName": {"Ref": "KeyName"},

```

```

      "InstanceType": "t3.micro",
      "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
      "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
      "UserData": {"Fn::Base64": {"Fn::Join": [ "", [
        "#!/bin/bash\n",
        "dnf update -y aws-cfn-bootstrap\n",
        "dnf install -y amazon-cloudwatch-agent\n",
        "/opt/aws/bin/cfn-init -v --stack ", {"Ref": "AWS::StackName"}, "
--resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n",
        "\n",
        "# Verify Apache log directory exists and create if needed\n",
        "mkdir -p /var/log/httpd\n",
        "\n",
        "# Start CloudWatch agent\n",
        "/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -
a fetch-config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json
-s\n",
        "\n",
        "# Signal success\n",
        "/opt/aws/bin/cfn-signal -e $? --stack ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n"
]]}}
    }
  },
  "WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "Properties": {
      "RetentionInDays": 7
    }
  },
  "404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
      "LogGroupName": {"Ref": "WebServerLogGroup"},
      "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code = 404, size, ...]",
      "MetricTransformations": [
        {
          "MetricValue": "1",
          "MetricNamespace": "test/404s",
          "MetricName": "test404Count"
        }
      ]
    }
  }
}

```

```

        ]
    },
    "BytesTransferredMetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {
            "LogGroupName": {"Ref": "WebServerLogGroup"},
            "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code, size, ...]",
            "MetricTransformations": [
                {
                    "MetricValue": "$size",
                    "MetricNamespace": "test/BytesTransferred",
                    "MetricName": "testBytesTransferred"
                }
            ]
        }
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "BandwidthAlarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The average volume of traffic is greater 3500 KB
over 10 minutes",
            "MetricName": "testBytesTransferred",
            "Namespace": "test/BytesTransferred",
            "Statistic": "Average",
            "Period": "300",
            "EvaluationPeriods": "2",
            "Threshold": "3500",

```

```

        "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
        "ComparisonOperator": "GreaterThanThreshold"
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {"Ref": "WebServerHost"}
        },
        "WebsiteURL": {
            "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
            "Description": "URL for the web server"
        },
        "PublicIP": {
            "Description": "Public IP address of the web server",
            "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
        },
        "CloudWatchLogGroupName": {
            "Description": "The name of the CloudWatch log group",
            "Value": {"Ref": "WebServerLogGroup"}
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Sample template that sets up and configures CloudWatch Logs on Amazon
Linux 2023 instance.
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.

```

SSHLocation:

Description: The IP address range that can be used to SSH to the EC2 instances

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

OperatorEmail:

Description: Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)

Type: String

Resources:**LogRole:**

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Principal:

Service:

- ec2.amazonaws.com

Action:

- 'sts:AssumeRole'

Path: /

Policies:

- PolicyName: LogRolePolicy

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

- 'logs:PutLogEvents'
- 'logs:DescribeLogStreams'
- 'logs:DescribeLogGroups'
- 'logs:CreateLogGroup'
- 'logs:CreateLogStream'

Resource: '*'

LogRoleInstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Path: /

Roles:

```

    - !Ref LogRole
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
WebServerHost:
  Type: AWS::EC2::Instance
  Metadata:
    Comment: Install a simple PHP application on Amazon Linux 2023
    'AWS::CloudFormation::Init':
      config:
        packages:
          dnf:
            httpd: []
            php: []
            php-fpm: []
        files:
          /etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json:
            content: !Sub |
              {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [
                        {
                          "file_path": "/var/log/httpd/access_log",
                          "log_group_name": "${WebServerLogGroup}",
                          "log_stream_name": "{instance_id}/apache.log",
                          "timestamp_format": "%d/%b/%Y:%H:%M:%S %z"
                        }
                      ]
                    }
                  }
                }
              }

```

```

        mode: '000644'
        owner: root
        group: root
/var/www/html/index.php:
        content: |
            <?php echo '<h1>AWS CloudFormation sample PHP application on Amazon
Linux 2023</h1>';
            ?>
        mode: '000644'
        owner: apache
        group: apache
/etc/cfn/cfn-hup.conf:
        content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
        mode: '000400'
        owner: root
        group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
        content: !Sub |
            [cfn-auto-reloader-hook]
            triggers=post.update
            path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerHost --
region ${AWS::Region}
            runas=root
    services:
        systemd:
            httpd:
                enabled: 'true'
                ensureRunning: 'true'
            php-fpm:
                enabled: 'true'
                ensureRunning: 'true'
    CreationPolicy:
        ResourceSignal:
            Timeout: PT5M
    Properties:
        ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
        KeyName: !Ref KeyName
        InstanceType: t3.micro
        SecurityGroupIds:

```



```

    - !Ref WebServerSecurityGroup
    IamInstanceProfile: !Ref LogRoleInstanceProfile
    UserData: !Base64
      Fn::Sub: |
        #!/bin/bash
        dnf update -y aws-cfn-bootstrap
        dnf install -y amazon-cloudwatch-agent
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServerHost
--region ${AWS::Region}

        # Verify Apache log directory exists and create if needed
        mkdir -p /var/log/httpd

        # Start CloudWatch agent
        /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json -s

        # Signal success
        /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerHost --region ${AWS::Region}
        echo "Done"
  WebServerLogGroup:
    Type: AWS::Logs::LogGroup
    DeletionPolicy: Retain
    UpdateReplacePolicy: Retain
    Properties:
      RetentionInDays: 7
  404MetricFilter:
    Type: AWS::Logs::MetricFilter
    Properties:
      LogGroupName: !Ref WebServerLogGroup
      FilterPattern: >-
        [ip, identity, user_id, timestamp, request, status_code = 404, size, ...]
      MetricTransformations:
        - MetricValue: '1'
          MetricNamespace: test/404s
          MetricName: test404Count
  BytesTransferredMetricFilter:
    Type: AWS::Logs::MetricFilter
    Properties:
      LogGroupName: !Ref WebServerLogGroup
      FilterPattern: '[ip, identity, user_id, timestamp, request, status_code,
size, ...]'
      MetricTransformations:

```

```
- MetricValue: $size
  MetricNamespace: test/BytesTransferred
  MetricName: testBytesTransferred
404Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
BandwidthAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The average volume of traffic is greater 3500 KB over 10
minutes
    MetricName: testBytesTransferred
    Namespace: test/BytesTransferred
    Statistic: Average
    Period: '300'
    EvaluationPeriods: '2'
    Threshold: '3500'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
AlarmNotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for the web server
  PublicIP:
```

```

Description: Public IP address of the web server
Value: !GetAtt WebServerHost.PublicIp
CloudWatchLogGroupName:
Description: The name of the CloudWatch log group
Value: !Ref WebServerLogGroup

```

Envoyer des journaux vers CloudWatch Logs à partir d'une instance Windows

Le modèle suivant configure les CloudWatch journaux pour une instance Windows 2012R2.

L'agent CloudWatch Logs sous Windows (agent SSM sous Windows 2012R2 et Windows 2016 AMIs) envoie des journaux uniquement après son démarrage, de sorte que les journaux générés avant le démarrage ne sont pas envoyés. Pour contourner ce problème, le modèle permet de s'assurer que l'agent démarre avant que les journaux ne soient écrits par :

- Configuration de l'agent défini comme premier config élément de cfn-init configSets.
- Utilisation de waitAfterCompletion pour insérer une pause après la commande qui démarre l'agent.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on
Windows 2012R2 instance.",
  "Parameters": {
    "KeyPair": {
      "Description": "Name of an existing EC2 KeyPair to enable RDP access to the
instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "RDPLocation": {
      "Description": "The IP address range that can be used to RDP to the EC2
instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",

```

```

        "ConstraintDescription": "must be a valid IP CIDR range of the form
x.x.x.x/x."
    },
    "OperatorEmail": {
        "Description": "Email address to notify when CloudWatch alarms are
triggered (404 errors)",
        "Type": "String"
    }
},
"Resources": {
    "WebServerSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable HTTP access via port 80 and RDP access via
port 3389",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": "80",
                    "ToPort": "80",
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": "3389",
                    "ToPort": "3389",
                    "CidrIp": {"Ref": "RDPLocation"}
                }
            ]
        }
    },
    "LogRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        }
                    }
                ]
            }
        }
    }
}

```

```

        "Action": [
            "sts:AssumeRole"
        ]
    },
    ],
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
    ],
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "LogRolePolicy",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "logs:Create*",
                            "logs:PutLogEvents",
                            "s3:GetObject"
                        ],
                        "Resource": [
                            "arn:aws:logs:*:*:*:*",
                            "arn:aws:s3:::*"
                        ]
                    }
                ]
            }
        }
    ],
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        }
    },
    "WebServerHost": {
        "Type": "AWS::EC2::Instance",
        "CreationPolicy": {

```

```

        "ResourceSignal": {
            "Timeout": "PT15M"
        }
    },
    "Metadata": {
        "AWS::CloudFormation::Init": {
            "configSets": {
                "config": [
                    "00-ConfigureCWLogs",
                    "01-InstallWebServer",
                    "02-ConfigureApplication",
                    "03-Finalize"
                ]
            },
            "00-ConfigureCWLogs": {
                "files": {
                    "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\\
\\AWS.EC2.Windows.CloudWatch.json": {
                        "content": {
                            "EngineConfiguration": {
                                "Components": [
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "ApplicationEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "Application"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SystemEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "System"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SecurityEventLog",
                                        "Parameters": {

```

```

        "Levels": "7",
        "LogName": "Security"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\Program

Files\\Amazon\\Ec2ConfigService\\Logs",

            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-

ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\cfn\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd

HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\inetpub\\

logs\\LogFiles\\W3SVC1",

```

```

        "TimeZoneKind": "UTC",
        "TimestampFormat": "yyyy-MM-dd

HH:mm:ss"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

ApplicationEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

SystemEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
}

```



```

    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
SecurityEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
EC2ConfigLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/
CfnInitLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",

```

```

        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

IISLogs",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
],
"Flows": {
    "Flows": [

"ApplicationEventLog,CloudWatchApplicationEventLog",

"SystemEventLog,CloudWatchSystemEventLog",

"SecurityEventLog,CloudWatchSecurityEventLog",

"EC2ConfigLog,CloudWatchEC2ConfigLog",
"CfnInitLog,CloudWatchCfnInitLog",
"IISLogs,CloudWatchIISLogs",
"MemoryPerformanceCounter,CloudWatch"

    ]
},
    "PollInterval": "00:00:05"
},
    "IsEnabled": true
}
    },
    "commands": {
        "0-enableSSM": {

```

```

        "command": "powershell.exe -Command \"Set-Service -Name
AmazonSSMAgent -StartupType Automatic\" ",
        "waitAfterCompletion": "0"
    },
    "1-restartSSM": {
        "command": "powershell.exe -Command \"Restart-Service
AmazonSSMAgent \"",
        "waitAfterCompletion": "30"
    }
},
"01-InstallWebServer": {
    "commands": {
        "01_install_webserver": {
            "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server -IncludeAllSubFeature\"",
            "waitAfterCompletion": "0"
        }
    }
},
"02-ConfigureApplication": {
    "files": {
        "c:\\\\Inetpub\\\\wwwroot\\\\index.htm": {
            "content": "<html> <head> <title>Test Application
Page</title> </head> <body> <h1>Congratulations!! Your IIS server is configured.</h1>
</body> </html>"
        }
    }
},
"03-Finalize": {
    "commands": {
        "00_signal_success": {
            "command": {
                "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
            },
            "waitAfterCompletion": "0"
        }
    }
}
},
"Properties": {
    "KeyName": {

```

```

        "Ref": "KeyPair"
    },
    "ImageId": "{{resolve:ssm:/aws/service/ami-windows-latest/
Windows_Server-2012-R2_RTM-English-64Bit-Base}}",
    "InstanceType": "t2.xlarge",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {
        "Fn::Base64": {
            "Fn::Join": [
                "",
                [
                    "<script>\n",
                    "wmic product where \"description='Amazon SSM Agent' \"
uninstall\n",
                    "wmic product where \"description='aws-cfn-bootstrap'
\" uninstall \n",
                    "start /wait c:\\\\Windows\\system32\\msiexec /passive /
qn /i https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi\n",
                    "powershell.exe -Command \"iwr https://
s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:\\\\AmazonSSMAgentSetup.exe\"\\n",
                    "start /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\n",
                    "cfn-init.exe -v -c config -s ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "
\n",
                    "</script>\n"
                ]
            ]
        }
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",
        "Properties": {
            "RetentionInDays": 7
        }
    },
    "404MetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {

```

```

        "LogGroupName": {"Ref": "LogGroup"},
        "FilterPattern": "[timestamps, serverip, method, uri, query, port,
dash, clientip, useragent, status_code = 404, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    }
},
"Outputs": {
    "InstanceId": {
        "Description": "The instance ID of the web server",
        "Value": {"Ref": "WebServerHost"}
    },
    "WebsiteURL": {
        "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
        "Description": "URL for the web server"
    }
},

```

```

    "PublicIP": {
      "Description": "Public IP address of the web server",
      "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
    },
    "CloudWatchLogGroupName": {
      "Description": "The name of the CloudWatch log group",
      "Value": {"Ref": "LogGroup"}
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch Logs on Windows 2012R2
  instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify when CloudWatch alarms are triggered (404
    errors)
    Type: String
Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'

```

```
    CidrIp: 0.0.0.0/0
  - IpProtocol: tcp
    FromPort: '3389'
    ToPort: '3389'
    CidrIp: !Ref RDPLocation
LogRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
  Path: /
  Policies:
    - PolicyName: LogRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 'logs:Create*'
              - 'logs:PutLogEvents'
              - 's3:GetObject'
            Resource:
              - 'arn:aws:logs:*:*:*'
              - 'arn:aws:s3::*:*'
LogRoleInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref LogRole
WebServerHost:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
```

```

Metadata:
  'AWS::CloudFormation::Init':
    configSets:
      config:
        - 00-ConfigureCWLogs
        - 01-InstallWebServer
        - 02-ConfigureApplication
        - 03-Finalize
    00-ConfigureCWLogs:
      files:
        'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
          content: !Sub |
            {
              "EngineConfiguration": {
                "Components": [
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "ApplicationEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Application"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SystemEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "System"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SecurityEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Security"
                    }
                  }
                ]
              }
            }

```



```

        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\Program Files\\Amazon\\
\\Ec2ConfigService\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\cfn\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\
\\W3SVC1",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E

```

```

        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/ApplicationEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SystemEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SecurityEventLog",
            "Region": "${AWS::Region}",

```

```

        "SecretKey": ""
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/EC2ConfigLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/CfnInitLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/IISLogs",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {

```

```

        "AccessKey": "",
        "NameSpace": "Windows/Default",
        "Region": "${AWS::Region}",
        "SecretKey": ""
    }
}
],
"Flows": {
    "Flows": [
        "ApplicationEventLog,CloudWatchApplicationEventLog",
        "SystemEventLog,CloudWatchSystemEventLog",
        "SecurityEventLog,CloudWatchSecurityEventLog",
        "EC2ConfigLog,CloudWatchEC2ConfigLog",
        "CfnInitLog,CloudWatchCfnInitLog",
        "IISLogs,CloudWatchIISLogs",
        "MemoryPerformanceCounter,CloudWatch"
    ]
},
"PollInterval": "00:00:05"
},
"IsEnabled": true
}
commands:
  0-enableSSM:
    command: >-
      powershell.exe -Command "Set-Service -Name AmazonSSMAgent
        -StartupType Automatic"
    waitAfterCompletion: '0'
  1-restartSSM:
    command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
    waitAfterCompletion: '30'
01-InstallWebServer:
  commands:
    01_install_webserver:
      command: >-
        powershell.exe -Command "Install-WindowsFeature Web-Server
          -IncludeAllSubFeature"
      waitAfterCompletion: '0'
02-ConfigureApplication:
  files:
    'c:\Inetpub\wwwroot\index.htm':
      content: >-
        <html> <head> <title>Test Application Page</title> </head>
        <body> <h1>Congratulations !! Your IIS server is

```

```

        configured.</h1> </body> </html>
    03-Finalize:
        commands:
            00_signal_success:
                command: !Sub >-
                    cfn-signal.exe -e 0 --resource WebServerHost --stack
                        ${AWS::StackName} --region ${AWS::Region}
                waitAfterCompletion: '0'
    Properties:
        KeyName: !Ref KeyPair
        ImageId: "{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2012-
R2_RTM-English-64Bit-Base}}"
        InstanceType: t2.xlarge
        SecurityGroupIds:
            - !Ref WebServerSecurityGroup
        IamInstanceProfile: !Ref LogRoleInstanceProfile
        UserData: !Base64
            'Fn::Sub': >
                <script>

                    wmic product where "description='Amazon SSM Agent' " uninstall

                    wmic product where "description='aws-cfn-bootstrap' " uninstall

                    start /wait c:\\Windows\\system32\\msiexec /passive /qn /i
                    https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi

                    powershell.exe -Command "iwr
                    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
                    -UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"

                    start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet

                    cfn-init.exe -v -c config -s ${AWS::StackName} --resource
                    WebServerHost --region ${AWS::Region}

                </script>
        LogGroup:
            Type: AWS::Logs::LogGroup
            Properties:
                RetentionInDays: 7
        404MetricFilter:

```

```
Type: AWS::Logs::MetricFilter
```

```
Properties:
```

```
  LogGroupName: !Ref LogGroup
```

```
  FilterPattern: >-
```

```
    [timestamps, serverip, method, uri, query, port, dash, clientip,  
    useragent, status_code = 404, ...]
```

```
  MetricTransformations:
```

```
    - MetricValue: '1'
```

```
      MetricNamespace: test/404s
```

```
      MetricName: test404Count
```

```
404Alarm:
```

```
Type: AWS::CloudWatch::Alarm
```

```
Properties:
```

```
  AlarmDescription: The number of 404s is greater than 2 over 2 minutes
```

```
  MetricName: test404Count
```

```
  Namespace: test/404s
```

```
  Statistic: Sum
```

```
  Period: '60'
```

```
  EvaluationPeriods: '2'
```

```
  Threshold: '2'
```

```
  AlarmActions:
```

```
    - !Ref AlarmNotificationTopic
```

```
  ComparisonOperator: GreaterThanThreshold
```

```
AlarmNotificationTopic:
```

```
Type: AWS::SNS::Topic
```

```
Properties:
```

```
  Subscription:
```

```
    - Endpoint: !Ref OperatorEmail
```

```
      Protocol: email
```

```
Outputs:
```

```
  InstanceId:
```

```
    Description: The instance ID of the web server
```

```
    Value: !Ref WebServerHost
```

```
  WebsiteURL:
```

```
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
```

```
    Description: URL for the web server
```

```
  PublicIP:
```

```
    Description: Public IP address of the web server
```

```
    Value: !GetAtt
```

```
      - WebServerHost
```

```
      - PublicIp
```

```
CloudWatchLogGroupName:
```

```
  Description: The name of the CloudWatch log group
```

Value: !Ref LogGroup

Consultez aussi

Pour plus d'informations sur CloudWatch les ressources Logs, consultez [AWS::Logs::LogGroup](#) ou [AWS::Logs::MetricFilter](#).

Extraits de modèle Amazon DynamoDB

Rubriques

- [Application Auto Scaling avec une table Amazon DynamoDB](#)
- [Consultez aussi](#)

Application Auto Scaling avec une table Amazon DynamoDB

Cet exemple configure Application Auto Scaling pour une ressource `AWS::DynamoDB::Table`. Le modèle définit une politique de dimensionnement `TargetTrackingScaling` qui adapte le débit `WriteCapacityUnits` à la hausse pour la table.

JSON

```
{
  "Resources": {
    "DDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [
          {
            "AttributeName": "ArtistId",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Concert",
            "AttributeType": "S"
          },
          {
            "AttributeName": "TicketSales",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
```

```

        {
            "AttributeName": "ArtistId",
            "KeyType": "HASH"
        },
        {
            "AttributeName": "Concert",
            "KeyType": "RANGE"
        }
    ],
    "GlobalSecondaryIndexes": [
        {
            "IndexName": "GSI",
            "KeySchema": [
                {
                    "AttributeName": "TicketSales",
                    "KeyType": "HASH"
                }
            ],
            "Projection": {
                "ProjectionType": "KEYS_ONLY"
            },
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            }
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
    }
},
"WriteCapacityScalableTarget": {
    "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties": {
        "MaxCapacity": 15,
        "MinCapacity": 5,
        "ResourceId": {
            "Fn::Join": [
                "/",
                [
                    "table",
                    {

```



```

        "Ref": "DDBTable"
      }
    ]
  ],
  "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
  "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
  "ServiceNamespace": "dynamodb"
}
},
"WriteScalingPolicy": {
  "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",
  "Properties": {
    "PolicyName": "WriteAutoScalingPolicy",
    "PolicyType": "TargetTrackingScaling",
    "ScalingTargetId": {
      "Ref": "WriteCapacityScalableTarget"
    },
    "TargetTrackingScalingPolicyConfiguration": {
      "TargetValue": 50,
      "ScaleInCooldown": 60,
      "ScaleOutCooldown": 60,
      "PredefinedMetricSpecification": {
        "PredefinedMetricType": "DynamoDBWriteCapacityUtilization"
      }
    }
  }
}
}
}
}

```

YAML

```
Resources:
  DDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
        - AttributeName: "ArtistId"
          AttributeType: "S"
        - AttributeName: "Concert"
```

```

        AttributeType: "S"
    - AttributeName: "TicketSales"
      AttributeType: "S"
  KeySchema:
    - AttributeName: "ArtistId"
      KeyType: "HASH"
    - AttributeName: "Concert"
      KeyType: "RANGE"
  GlobalSecondaryIndexes:
    - IndexName: "GSI"
      KeySchema:
        - AttributeName: "TicketSales"
          KeyType: "HASH"
      Projection:
        ProjectionType: "KEYS_ONLY"
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      ResourceId: !Join
        - /
        - - table
          - !Ref DDBTable
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ServiceNamespace: dynamodb
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteAutoScalingPolicy
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0

```

```
ScaleInCooldown: 60
ScaleOutCooldown: 60
PredefinedMetricSpecification:
  PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Consultez aussi

Pour plus d'informations, consultez le billet de blog [Comment utiliser CloudFormation pour configurer le dimensionnement automatique pour les tables et les index DynamoDB](#) sur le blog de base de données. AWS

Pour plus d'informations sur les ressources DynamoDB, consultez. [AWS::DynamoDB::Table](#)

Extraits EC2 CloudFormation de modèles Amazon

Amazon EC2 fournit une capacité de calcul évolutive dans le AWS Cloud. Vous pouvez utiliser Amazon EC2 pour lancer autant ou aussi peu de serveurs virtuels que nécessaire, configurer la sécurité et le réseau, et gérer le stockage. Ces serveurs virtuels, appelés instances, peuvent exécuter divers systèmes d'exploitation et applications et peuvent être personnalisés pour répondre à vos besoins spécifiques. Amazon vous EC2 permet d'augmenter ou de réduire votre taille pour faire face à l'évolution des exigences ou aux pics d'utilisation.

Vous pouvez définir et approvisionner des EC2 instances Amazon dans le cadre de votre infrastructure à l'aide CloudFormation de modèles. Les modèles facilitent la gestion et l'automatisation du déploiement des EC2 ressources Amazon de manière reproductible et cohérente.

Les exemples d'extraits de modèle suivants décrivent CloudFormation des ressources ou des composants pour Amazon. EC2 Ces extraits sont conçus pour être intégrés dans un modèle et ne sont pas destinés à être exécutés indépendamment.

Catégories d'extraits

- [Configurez les instances Amazon EC2 avec CloudFormation](#)
- [Créez des modèles de lancement avec CloudFormation](#)
- [Gérez les groupes de sécurité avec CloudFormation](#)
- [Allouez et associez des adresses IP Elastic à CloudFormation](#)
- [Configurez les ressources Amazon VPC avec CloudFormation](#)

Configurez les instances Amazon EC2 avec CloudFormation

Les extraits suivants montrent comment configurer les instances Amazon EC2 à l'aide d'CloudFormation.

Catégories d'extraits

- [Configurations générales d'Amazon EC2](#)
- [Spécifier les mappages de périphérique de stockage en mode bloc pour une instance](#)

Configurations générales d'Amazon EC2

Les extraits suivants illustrent les configurations générales pour les instances Amazon EC2 utilisant CloudFormation.

Exemples d'extraits

- [Créer une instance Amazon EC2 dans une zone de disponibilité donnée](#)
- [Configurer une instance Amazon EC2 balisée avec un volume EBS et des données utilisateur](#)
- [Définir le nom de la table DynamoDB dans les données utilisateur pour le lancement de l'instance Amazon EC2](#)
- [Créer un volume Amazon EBS avec DeletionPolicy](#)

Créer une instance Amazon EC2 dans une zone de disponibilité donnée

L'extrait suivant crée une instance Amazon EC2 dans la zone de disponibilité spécifiée à l'aide d'une ressource. [AWS::EC2::Instance](#) Le code d'une zone de disponibilité représente son code de région suivi d'un identifiant à lettre. Vous pouvez lancer une instance dans une seule zone de disponibilité.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "AvailabilityZone": "aa-example-1a",
    "ImageId": "ami-1234567890abcdef0"
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
```

Configurer une instance Amazon EC2 balisée avec un volume EBS et des données utilisateur

L'extrait suivant crée une instance Amazon EC2 avec une balise, un volume EBS et des données utilisateur. Il utilise une [AWS::EC2::Instance](#) ressource. Dans le même modèle, vous devez définir une [AWS::EC2::SecurityGroup](#) ressource, une [AWS::SNS::Topic](#) ressource et une [AWS::EC2::Volume](#) ressource. Le KeyName doit être défini dans la section du modèle `Parameters`.

Les balises peuvent vous aider à classer les AWS ressources en fonction de vos préférences, par exemple par objectif, propriétaire ou environnement. Les données utilisateur permettent de fournir des scripts ou des données personnalisés à une instance lors du lancement. Ces données facilitent l'automatisation des tâches, la configuration logicielle, l'installation de packages et d'autres actions sur une instance lors de l'initialisation.

Pour plus d'informations sur le marquage de vos ressources, consultez la section [Marquer vos ressources Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

Pour plus d'informations sur les données utilisateur, consultez la section [Utiliser les métadonnées d'instance pour gérer votre instance EC2](#) dans le Guide de l'utilisateur Amazon EC2.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups": [ { "Ref": "Ec2SecurityGroup" } ],
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [ ":", [
          "PORT=80",
          "TOPIC=",
          { "Ref": "MySNSTopic" }
        ] ]
      }
    }
  }
}
```

```

    ]
  }
},
"InstanceType": "aa.size",
"AvailabilityZone": "aa-example-1a",
"ImageId": "ami-1234567890abcdef0",
"Volumes": [
  {
    "VolumeId": { "Ref": "MyVolumeResource" },
    "Device": "/dev/sdk"
  }
],
"Tags": [ { "Key": "Name", "Value": "MyTag" } ]
}
}

```

YAML

```

Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    KeyName: !Ref KeyName
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    UserData:
      Fn::Base64:
        Fn::Join:
          - ":"
          - - "PORT=80"
            - "TOPIC="
            - !Ref MySNSTopic
    InstanceType: aa.size
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
    Volumes:
      - VolumeId: !Ref MyVolumeResource
        Device: "/dev/sdk"
    Tags:
      - Key: Name
        Value: MyTag

```

Définir le nom de la table DynamoDB dans les données utilisateur pour le lancement de l'instance Amazon EC2

L'extrait suivant crée une instance Amazon EC2 et définit un nom de table DynamoDB dans les données utilisateur à transmettre à l'instance lors du lancement. Il utilise une [AWS::EC2::Instance](#) ressource. Vous pouvez définir des paramètres ou des valeurs dynamiques dans les données utilisateur pour transmettre une instance EC2 au lancement.

Pour plus d'informations sur les données utilisateur, consultez la section [Utiliser les métadonnées d'instance pour gérer votre instance EC2](#) dans le Guide de l'utilisateur Amazon EC2.

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [
          "",
          [
            "TableName=",
            {
              "Ref": "DynamoDBTableName"
            }
          ]
        ]
      },
      "AvailabilityZone": "aa-example-1a",
      "ImageId": "ami-1234567890abcdef0"
    }
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    UserData:
      Fn::Base64:
        Fn::Join:
```

```

- ''
- - 'TableName='
  - Ref: DynamoDBTableName
AvailabilityZone: aa-example-1a
ImageId: ami-1234567890abcdef0

```

Créer un volume Amazon EBS avec **DeletionPolicy**

Les extraits suivants créent un volume Amazon EBS à l'aide d'une ressource Amazon EC2.

[AWS::EC2::Volume](#) Vous pouvez utiliser les propriétés `Size` or `SnapshotID` pour définir le volume, mais pas les deux. Un attribut `DeletionPolicy` est défini pour créer un instantané du volume lorsque la pile est supprimée.

Pour plus d'informations sur l'attribut `DeletionPolicy`, consultez [Attribut DeletionPolicy](#).

Pour plus d'informations sur la création d'es volumes Amazon EBS, consultez [Créer un volume Amazon EBS](#).

JSON

Cet extrait crée un volume Amazon EBS d'une taille spécifiée. La taille est fixée à 10, mais vous pouvez l'ajuster selon vos besoins. La [AWS::EC2::Volume](#) ressource vous permet de spécifier la taille ou un identifiant de capture d'écran, mais pas les deux.

```

"MyEBSVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "Size": "10",
    "AvailabilityZone": {
      "Ref": "AvailabilityZone"
    }
  },
  "DeletionPolicy": "Snapshot"
}

```

Cet extrait crée un volume Amazon EBS à l'aide d'un ID d'instantané fourni. La [AWS::EC2::Volume](#) ressource vous permet de spécifier la taille ou un identifiant de capture d'écran, mais pas les deux.

```

"MyEBSVolume": {

```



```
"Type": "AWS::EC2::Volume",
"Properties": {
  "SnapshotId" : "snap-1234567890abcdef0",
  "AvailabilityZone": {
    "Ref": "AvailabilityZone"
  }
},
"DeletionPolicy": "Snapshot"
}
```

YAML

Cet extrait crée un volume Amazon EBS d'une taille spécifiée. La taille est fixée à 10, mais vous pouvez l'ajuster selon vos besoins. La [AWS::EC2::Volume](#) ressource vous permet de spécifier la taille ou un identifiant de capture d'écran, mais pas les deux.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Cet extrait crée un volume Amazon EBS à l'aide d'un ID d'instantané fourni. La [AWS::EC2::Volume](#) ressource vous permet de spécifier la taille ou un identifiant de capture d'écran, mais pas les deux.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    SnapshotId: snap-1234567890abcdef0
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Spécifier les mappages de périphérique de stockage en mode bloc pour une instance

Un mappage de périphérique de stockage en mode bloc définit les périphériques de stockage en mode bloc qui comprennent les volumes de stockage d'instance et volumes EBS devant être attachés à l'instance. Vous pouvez spécifier un mappage de périphérique de stockage en mode bloc

lors de la création d'une AMI, afin que le mappage soit utilisé par toutes les instances lancées à partir de l'AMI. Vous pouvez également spécifier un mappage de périphérique de stockage en mode bloc lorsque vous lancez une instance, afin que son mappage remplace celui spécifié dans l'AMI à partir de laquelle vous avez lancé l'instance.

Vous pouvez utiliser les extraits de modèle suivants pour spécifier les mappages de périphériques par blocs pour vos volumes EBS ou de stockage d'instance à l'aide de la `BlockDeviceMappings` propriété d'une ressource. [AWS::EC2::Instance](#)

Pour plus d'informations sur les mappages de périphériques de stockage en mode bloc, consultez la section [Mappages de périphériques de stockage en mode bloc pour les volumes sur les instances Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

Scénarios

- [Spécifier les mappages de périphériques de stockage en mode bloc pour deux volumes EBS](#)
- [Spécifier le mappage de périphérique de stockage en mode bloc d'un volume de stockage d'instances](#)

Spécifier les mappages de périphériques de stockage en mode bloc pour deux volumes EBS

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "KeyName": { "Ref": "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/sda1",
        "Ebs": { "VolumeSize": "50" }
      },
      {
        "DeviceName": "/dev/sdm",
        "Ebs": { "VolumeSize": "100" }
      }
    ]
  }
}
```

```

    }
  }
}
```

YAML

```

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 50
      -
        DeviceName: /dev/sdm
        Ebs:
          VolumeSize: 100
```

Spécifier le mappage de périphérique de stockage en mode bloc d'un volume de stockage d'instances

JSON

```

"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [ { "Ref" : "Ec2SecurityGroup" } ],
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdc",
        "VirtualName" : "ephemeral0"
```

```
    }  
  ]  
}  
}
```

YAML

```
EC2Instance:  
  Type: AWS::EC2::Instance  
  Properties:  
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-  
gp2}}'  
    KeyName: !Ref KeyName  
    InstanceType: !Ref InstanceType  
    SecurityGroups:  
      - !Ref Ec2SecurityGroup  
    BlockDeviceMappings:  
      - DeviceName: /dev/sdc  
        VirtualName: ephemeral0
```

Créez des modèles de lancement avec CloudFormation

Cette section fournit un exemple de création d'un modèle de lancement Amazon EC2 à l'aide de CloudFormation. Les modèles de lancement vous permettent de créer des modèles pour configurer et approvisionner des instances Amazon EC2 au sein de AWS. Les modèles de lancement vous permettent de stocker les paramètres de lancement afin de ne pas avoir à les spécifier à chaque fois que vous lancez une instance. Pour plus d'exemples, consultez la section [Exemples](#) dans la ressource `AWS::EC2::LaunchTemplate`.

Pour plus d'informations sur les modèles de lancement, consultez la section [Paramètres de lancement d'instance de magasin dans les modèles de lancement Amazon EC2](#) du Guide de l'utilisateur Amazon EC2.

Pour plus d'informations sur la création de modèles de lancement à utiliser avec les groupes Auto Scaling, consultez la section [Modèles de lancement autoscaling](#) dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Catégories d'extraits

- [Création d'un modèle de lancement qui spécifie les groupes de sécurité, les balises, les données utilisateur et un rôle IAM](#)

Création d'un modèle de lancement qui spécifie les groupes de sécurité, les balises, les données utilisateur et un rôle IAM

Cet extrait montre une [AWS::EC2::LaunchTemplate](#) ressource contenant les informations de configuration nécessaires au lancement d'une instance. Vous spécifiez les valeurs pour les propriétés ImageId, InstanceType, SecurityGroups, UserData et TagSpecifications. La propriété SecurityGroups indique un groupe de sécurité EC2 existant nommé et un nouveau groupe de sécurité. La Ref fonction obtient l'ID de la [AWS::EC2::SecurityGroup](#) ressource myNewEC2SecurityGroup déclarée ailleurs dans le modèle de pile.

Le modèle de lancement comprend une section pour les données utilisateur personnalisées. Dans cette section, vous pouvez transmettre des tâches de configuration et des scripts qui s'exécutent lors du lancement d'une instance. Dans cet exemple, les données utilisateur installent l'AWS Systems Manager agent et démarrent l'agent.

Le modèle de lancement inclut également un rôle IAM qui autorise les applications exécutées sur les instances à effectuer des actions en votre nom. Cet exemple montre une [AWS::IAM::Role](#) ressource pour le modèle de lancement, qui utilise la IamInstanceProfile propriété pour spécifier le rôle IAM. La Ref fonction obtient le nom de la [AWS::IAM::InstanceProfile](#) ressource myInstanceProfile. Pour configurer les autorisations du rôle IAM, vous spécifiez une valeur pour la propriété ManagedPolicyArns.

JSON

```
{
  "Resources":{
    "myLaunchTemplate":{
      "Type":"AWS::EC2::LaunchTemplate",
      "Properties":{
        "LaunchTemplateName":{ "Fn::Sub": "${AWS::StackName}-launch-template" },
        "LaunchTemplateData":{
          "ImageId":"ami-02354e95b3example",
          "InstanceType":"t3.micro",
          "IamInstanceProfile":{
            "Name":{
              "Ref":"myInstanceProfile"
            }
          },
        },
        "SecurityGroupIds":[
          {
            "Ref":"myNewEC2SecurityGroup"
          }
        ]
      }
    }
  }
}
```

```

    },
    "sg-083cd3bfb8example"
  ],
  "UserData":{
    "Fn::Base64":{
      "Fn::Join": [
        "", [
          "#!/bin/bash\n",
          "cd /tmp\n",
          "yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm\n",
          "systemctl enable amazon-ssm-agent\n",
          "systemctl start amazon-ssm-agent\n"
        ]
      ]
    }
  },
  "TagSpecifications":[
    {
      "ResourceType":"instance",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    },
    {
      "ResourceType":"volume",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    }
  ]
},
"myInstanceRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "RoleName":"InstanceRole",

```

```

    "AssumeRolePolicyDocument":{
      "Version": "2012-10-17",
      "Statement":[
        {
          "Effect":"Allow",
          "Principal":{"
            "Service":[
              "ec2.amazonaws.com"
            ]
          },
          "Action":[
            "sts:AssumeRole"
          ]
        }
      ]
    },
    "ManagedPolicyArns":[
      "arn:aws:iam::aws:policy/myCustomerManagedPolicy"
    ]
  }
},
"myInstanceProfile":{
  "Type":"AWS::IAM::InstanceProfile",
  "Properties":{
    "Path":"/",
    "Roles":[
      {
        "Ref":"myInstanceRole"
      }
    ]
  }
}
}
}
}
}

```

YAML

```

---
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template

```

```

LaunchTemplateData:
  ImageId: ami-02354e95b3example
  InstanceType: t3.micro
  IamInstanceProfile:
    Name: !Ref myInstanceProfile
  SecurityGroupIds:
    - !Ref myNewEC2SecurityGroup
    - sg-083cd3bfb8example
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash
      cd /tmp
      yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
      systemctl enable amazon-ssm-agent
      systemctl start amazon-ssm-agent
  TagSpecifications:
    - ResourceType: instance
      Tags:
        - Key: environment
          Value: development
    - ResourceType: volume
      Tags:
        - Key: environment
          Value: development
myInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: InstanceRole
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: 'Allow'
          Principal:
            Service:
              - 'ec2.amazonaws.com'
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/myCustomerManagedPolicy'
myInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: '/'

```



```
Roles:
- !Ref myInstanceRole
```

Gérez les groupes de sécurité avec CloudFormation

Les extraits suivants montrent comment gérer les groupes de sécurité et les instances Amazon EC2 CloudFormation afin de contrôler l'accès à vos ressources. AWS

Catégories d'extraits

- [Associer une instance Amazon EC2 à un groupe de sécurité](#)
- [Créer des groupes de sécurité avec des règles d'entrée](#)
- [Créer un Elastic Load Balancer avec une règle d'entrée de groupe de sécurité](#)

Associer une instance Amazon EC2 à un groupe de sécurité

Les exemples d'extraits suivants montrent comment associer une instance Amazon EC2 à un groupe de sécurité Amazon VPC par défaut à l'aide d' CloudFormation.

Exemples d'extraits

- [Associer une instance Amazon EC2 à un groupe de sécurité VPC par défaut](#)
- [Créer une instance Amazon EC2 avec un volume et un groupe de sécurité attachés](#)

Associer une instance Amazon EC2 à un groupe de sécurité VPC par défaut

L'extrait suivant crée un Amazon VPC, un sous-réseau au sein du VPC et une instance Amazon EC2. Le VPC est créé à l'aide d'une ressource [AWS::EC2::VPC](#). La plage d'adresses IP du VPC est définie dans le modèle plus large et est référencée par le paramètre MyVPCCIDRRange.

Un sous-réseau est créé au sein du VPC à l'aide d'une ressource [AWS::EC2::Subnet](#). Le sous-réseau est associé au VPC, qui est référencé comme MyVPC.

Une instance EC2 est lancée au sein du VPC et du sous-réseau à l'aide d'une ressource.

[AWS::EC2::Instance](#) Cette ressource indique l'Amazon Machine Image (AMI) à utiliser pour lancer l'instance, le sous-réseau sur lequel l'instance sera exécutée et le groupe de sécurité à associer à l'instance. Le ImageId utilise un paramètre Systems Manager pour récupérer dynamiquement la dernière AMI Amazon Linux 2.

L'ID du groupe de sécurité est obtenu à l'aide de la fonction `Fn::GetAtt`, qui extrait le groupe de sécurité par défaut de la ressource `MyVPC`.

L'instance est placée dans la ressource `MySubnet` définie dans l'extrait.

Lorsque vous créez un VPC à l'aide de CloudFormation, crée AWS automatiquement des ressources par défaut au sein du VPC, y compris un groupe de sécurité par défaut. Toutefois, lorsque vous définissez un VPC dans un CloudFormation modèle, il se peut que vous n'ayez pas accès à ces ressources par défaut lors IDs de la création du modèle. Pour accéder aux ressources par défaut spécifiées dans le modèle et les utiliser, vous pouvez utiliser des fonctions intrinsèques telles que `Fn::GetAtt`. Cette fonction vous permet de travailler avec les ressources par défaut créées automatiquement par CloudFormation.

JSON

```
"MyVPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "EnableDnsSupport": false,
    "EnableDnsHostnames": false,
    "InstanceTenancy": "default"
  }
},
"MySubnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "VpcId": {
      "Ref": "MyVPC"
    }
  }
},
"MyInstance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
```

```

    "SecurityGroupIds": [
      {
        "Fn::GetAtt": [
          "MyVPC",
          "DefaultSecurityGroup"
        ]
      }
    ],
    "SubnetId": {
      "Ref": "MySubnet"
    }
  }
}

```

YAML

```

MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}'
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet

```

Créer une instance Amazon EC2 avec un volume et un groupe de sécurité attachés

L'extrait suivant crée une instance Amazon EC2 à l'aide d'[AWS::EC2::Instance](#) une ressource, qui est lancée à partir d'une AMI désignée. L'instance est associée à un groupe de sécurité qui autorise le trafic SSH entrant sur le port 22 à partir d'une adresse IP spécifiée, à l'aide d'une [AWS::EC2::SecurityGroup](#) ressource. Il crée un volume Amazon EBS de 100 Go à l'aide d'une [AWS::EC2::Volume](#) ressource. Le volume est créé dans la même zone de disponibilité que l'instance, telle que spécifiée par la fonction `GetAtt`, et est monté sur l'instance sur le périphérique `/dev/sdh`.

Pour plus d'informations sur la création d'es volumes Amazon EBS, consultez [Créer un volume Amazon EBS](#).

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "SecurityGroups": [
      {
        "Ref": "InstanceSecurityGroup"
      }
    ],
    "ImageId": "ami-1234567890abcdef0"
  }
},
"InstanceSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Enable SSH access via port 22",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
      }
    ]
  }
},
"NewVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "Size": "100",
```

```

        "AvailabilityZone": {
            "Fn::GetAtt": [
                "Ec2Instance",
                "AvailabilityZone"
            ]
        }
    },
    "MountPoint": {
        "Type": "AWS::EC2::VolumeAttachment",
        "Properties": {
            "InstanceId": {
                "Ref": "Ec2Instance"
            },
            "VolumeId": {
                "Ref": "NewVolume"
            },
            "Device": "/dev/sdh"
        }
    }
}

```

YAML

```

Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    SecurityGroups:
      - !Ref InstanceSecurityGroup
    ImageId: ami-1234567890abcdef0
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24
NewVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt [Ec2Instance, AvailabilityZone]

```

```
MountPoint:
  Type: AWS::EC2::VolumeAttachment
  Properties:
    InstanceId: !Ref Ec2Instance
    VolumeId: !Ref NewVolume
    Device: /dev/sdh
```

Créer des groupes de sécurité avec des règles d'entrée

Les exemples d'extraits suivants montrent comment configurer des groupes de sécurité avec des règles d'entrée spécifiques à l'aide d' CloudFormation.

Extraits

- [Créer un groupe de sécurité avec des règles d'entrée pour l'accès SSH et HTTP](#)
- [Créer un groupe de sécurité avec des règles d'entrée pour l'accès HTTP et SSH à partir des plages d'adresses CIDR spécifiées](#)
- [Créer des groupes de sécurité à référencement croisé avec des règles d'entrée](#)

Créer un groupe de sécurité avec des règles d'entrée pour l'accès SSH et HTTP

L'extrait suivant décrit deux règles d'entrée de groupes de sécurité utilisant une ressource.

[AWS::EC2::SecurityGroup](#) La première règle d'entrée autorise l'accès SSH (port 22) à partir d'un groupe de sécurité existant nommé MyAdminSecurityGroup, qui appartient au AWS compte ayant le numéro de compte. 1111-2222-3333 La deuxième règle d'entrée autorise l'accès HTTP (port 80) à partir d'un groupe de sécurité différent nommé MySecurityGroupCreatedInCFN, créé dans le même modèle. La fonction Ref est utilisée pour référencer le nom logique du groupe de sécurité créé dans le même modèle.

Dans la première règle d'entrée, vous devez ajouter une valeur à la fois pour les propriétés SourceSecurityGroupName et SourceSecurityGroupOwnerId. Dans la deuxième règle d'entrée, MySecurityGroupCreatedInCFNTemplate fait référence à un autre groupe de sécurité créé dans le même modèle. Vérifiez que le nom logique MySecurityGroupCreatedInCFNTemplate correspond au nom logique réel de la ressource de groupe de sécurité que vous spécifiez dans le modèle plus large.

Pour plus d'informations sur les groupes de sécurité, consultez la section [Groupes de sécurité Amazon EC2 pour vos instances Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

JSON

```
"SecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified source security group",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "SourceSecurityGroupName": "MyAdminSecurityGroup",
        "SourceSecurityGroupOwnerId": "1111-2222-3333"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "SourceSecurityGroupName": {
          "Ref": "MySecurityGroupCreatedInCFNTemplate"
        }
      }
    ]
  }
}
```

YAML

```
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified source security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        SourceSecurityGroupName: MyAdminSecurityGroup
        SourceSecurityGroupOwnerId: '1111-2222-3333'
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        SourceSecurityGroupName:
          Ref: MySecurityGroupCreatedInCFNTemplate
```

Créer un groupe de sécurité avec des règles d'entrée pour l'accès HTTP et SSH à partir des plages d'adresses CIDR spécifiées

L'extrait suivant crée un groupe de sécurité pour une instance Amazon EC2 avec deux règles entrantes. Les règles entrantes autorisent le trafic TCP entrant sur les ports spécifiés à partir des plages d'adresses CIDR désignées. Une [AWS::EC2::SecurityGroup](#) ressource est utilisée pour définir les règles. Vous devez spécifier un protocole pour chaque règle. Pour TCP, vous devez spécifier un port ou la plage de ports. Si vous ne spécifiez ni le groupe de sécurité source ni de plage d'adresses CIDR, la pile est lancée avec succès, mais la règle n'est pas appliquée au groupe de sécurité.

Pour plus d'informations sur les groupes de sécurité, consultez la section [Groupes de sécurité Amazon EC2 pour vos instances Amazon EC2](#) dans le Guide de l'utilisateur Amazon EC2.

JSON

```
"ServerSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified CIDR ranges",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "CidrIp": "192.0.2.0/24"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
      }
    ]
  }
}
```

YAML

```
ServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified CIDR ranges
```



```
SecurityGroupIngress:
  - IpProtocol: tcp
    FromPort: 80
    ToPort: 80
    CidrIp: 192.0.2.0/24
  - IpProtocol: tcp
    FromPort: 22
    ToPort: 22
    CidrIp: 192.0.2.0/24
```

Créer des groupes de sécurité à référencement croisé avec des règles d'entrée

L'extrait de code suivant utilise la [AWS::EC2::SecurityGroup](#) ressource pour créer deux groupes de sécurité Amazon EC2, et. SGroup1 SGroup2 Les règles d'entrée qui autorisent la communication entre les deux groupes de sécurité sont créées à l'aide de la ressource [AWS::EC2::SecurityGroupIngress](#). SGroup1Ingress établit une règle d'entrée SGroup1 qui autorise le trafic TCP entrant sur le port 80 en provenance du groupe de sécurité source,. SGroup2 SGroup2Ingress établit une règle d'entrée SGroup2 qui autorise le trafic TCP entrant sur le port 80 en provenance du groupe de sécurité source,. SGroup1

JSON

```
"SGroup1": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "EC2 instance access"
  }
},
"SGroup2": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "EC2 instance access"
  }
},
"SGroup1Ingress": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties": {
    "GroupName": {
      "Ref": "SGroup1"
    },
    "IpProtocol": "tcp",
    "ToPort": "80",
```

```

        "FromPort": "80",
        "SourceSecurityGroupName": {
            "Ref": "SGroup2"
        }
    },
    "SGroup2Ingress": {
        "Type": "AWS::EC2::SecurityGroupIngress",
        "Properties": {
            "GroupName": {
                "Ref": "SGroup2"
            },
            "IpProtocol": "tcp",
            "ToPort": "80",
            "FromPort": "80",
            "SourceSecurityGroupName": {
                "Ref": "SGroup1"
            }
        }
    }
}

```

YAML

```

SGroup1:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup2:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup1Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupName: !Ref SGroup1
    IpProtocol: tcp
    ToPort: 80
    FromPort: 80
    SourceSecurityGroupName: !Ref SGroup2
SGroup2Ingress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupName: !Ref SGroup2

```

```
IpProtocol: tcp
ToPort: 80
FromPort: 80
SourceSecurityGroupName: !Ref SGroup1
```

Créer un Elastic Load Balancer avec une règle d'entrée de groupe de sécurité

Le modèle suivant crée une [AWS::ElasticLoadBalancing::LoadBalancer](#) ressource dans la zone de disponibilité spécifiée. La [AWS::ElasticLoadBalancing::LoadBalancer](#) ressource est configurée pour écouter le trafic HTTP sur le port 80 et pour diriger les demandes vers les instances également sur le port 80. L'Elastic Load Balancer est chargé d'équilibrer la charge du trafic HTTP entrant entre les instances.

En outre, ce modèle génère une [AWS::EC2::SecurityGroup](#) ressource associée à l'équilibreur de charge. Ce groupe de sécurité est créé avec une règle d'entrée unique, décrite comme `ELB ingress group`, qui autorise le trafic TCP entrant sur le port 80. La source de cette règle d'entrée est définie à l'aide de la fonction `Fn::GetAtt` de récupération des attributs de la ressource d'équilibrage de charge. `SourceSecurityGroupOwnerId` utilise `Fn::GetAtt` pour obtenir le `OwnerAlias` du groupe de sécurité source de l'équilibreur de charge. `SourceSecurityGroupName` utilise `Fn::GetAtt` pour obtenir le `GroupName` du groupe de sécurité source de l'ELB.

Cette configuration garantit une communication sécurisée entre l'ELB et les instances.

Pour plus d'informations sur l'équilibrage de charge, consultez [Guide de l'utilisateur Elastic Load Balancing](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyELB": {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {
        "AvailabilityZones": [
          "aa-example-1a"
        ],
        "Listeners": [
          {
            "LoadBalancerPort": "80",
```

```

        "InstancePort": "80",
        "Protocol": "HTTP"
      }
    ]
  },
  "MyELBIngressGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "ELB ingress group",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "SourceSecurityGroupOwnerId": {
            "Fn::GetAtt": [
              "MyELB",
              "SourceSecurityGroup.OwnerAlias"
            ]
          },
          "SourceSecurityGroupName": {
            "Fn::GetAtt": [
              "MyELB",
              "SourceSecurityGroup.GroupName"
            ]
          }
        }
      ]
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyELB:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
      AvailabilityZones:
        - aa-example-1a

```

```
Listeners:
  - LoadBalancerPort: '80'
    InstancePort: '80'
    Protocol: HTTP
MyELBIngressGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: ELB ingress group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        SourceSecurityGroupOwnerId:
          Fn::GetAtt:
            - MyELB
            - SourceSecurityGroup.OwnerAlias
    SourceSecurityGroupName:
      Fn::GetAtt:
        - MyELB
        - SourceSecurityGroup.GroupName
```

Allouez et associez des adresses IP Elastic à CloudFormation

Les extraits de modèle suivants sont des exemples relatifs aux adresses IP Elastic (EIP) dans Amazon EC2. Ces exemples couvrent l'allocation, l'association et la gestion EIPs de vos instances.

Exemples d'extraits

- [Attribuer une adresse IP Elastic et l'associer à une instance Amazon EC2](#)
- [Associer une adresse IP Elastic à une instance Amazon EC2 en spécifiant l'adresse IP](#)
- [Associer une adresse IP Elastic à une instance Amazon EC2 en spécifiant l'ID d'allocation de l'adresse IP](#)

Attribuer une adresse IP Elastic et l'associer à une instance Amazon EC2

L'extrait suivant alloue une adresse IP élastique (EIP) Amazon EC2 et l'associe à une instance Amazon EC2 à l'aide d'une ressource. [AWS::EC2::EIP](#) Vous pouvez allouer une adresse EIP à partir d'un pool d'adresses appartenant à AWS ou à partir d'un pool d'adresses créé à partir d'une plage d'IPv4 adresses publiques que vous utilisez AWS pour vos AWS ressources à l'aide de [Bring your own IP addresses \(BYOIP\)](#). Dans cet exemple, l'EIP est alloué à partir d'un pool d'adresses appartenant AWS.

Pour plus d'informations sur les adresses IP Elastic, consultez [Adresses IP Elastic](#) dans le Guide de l'utilisateur Amazon EC2.

JSON

```
"ElasticIP": {
  "Type": "AWS::EC2::EIP",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    }
  }
}
```

YAML

```
ElasticIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref EC2Instance
```

Associer une adresse IP Elastic à une instance Amazon EC2 en spécifiant l'adresse IP

L'extrait suivant associe une adresse IP élastique Amazon EC2 existante à une instance EC2 à l'aide d'une ressource [AWS::EC2::EIPAssociation](#). Vous devez d'abord attribuer une adresse IP Elastic à utiliser dans votre compte. Une adresse IP Elastic peut être associée à une seule instance.

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "EIP": "192.0.2.0"
  }
}
```

YAML

```
IPAssoc:
```

```
Type: AWS::EC2::EIPAssociation
Properties:
  InstanceId: !Ref EC2Instance
  EIP: 192.0.2.0
```

Associer une adresse IP Elastic à une instance Amazon EC2 en spécifiant l'ID d'allocation de l'adresse IP

L'extrait suivant associe une adresse IP élastique existante à une instance Amazon EC2 en spécifiant l'ID d'allocation à l'aide d'une ressource [AWS::EC2::EIPAssociation](#). Un ID d'allocation est attribué à une adresse IP Elastic lors de l'allocation d'adresse IP élastique.

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "AllocationId": "eipalloc-1234567890abcdef0"
  }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    AllocationId: eipalloc-1234567890abcdef0
```

Configurez les ressources Amazon VPC avec CloudFormation

Cette section fournit des exemples de configuration des ressources Amazon VPC à l'aide de CloudFormation. VPCs vous permettent de créer un réseau virtuel à l'intérieur AWS, et ces extraits montrent comment configurer certains aspects de pour répondre VPCs à vos exigences réseau.

Exemples d'extraits

- [Activer l'accès Internet IPv6 de sortie uniquement dans un VPC](#)
- [Extraits de modèle d'interface réseau Elastic \(ENI\)](#)

Activer l'accès Internet IPv6 de sortie uniquement dans un VPC

Une passerelle Internet de sortie uniquement permet aux instances d'un VPC d'accéder à Internet et d'empêcher les ressources sur Internet de communiquer avec les instances. L'extrait suivant active l'accès à Internet en IPv6 sortie uniquement depuis un VPC. Il crée un VPC avec une plage d'IPv4 adresses `10.0.0/16` utilisant une ressource [AWS::EC2::VPC](#). Une table de routage est associée à cette ressource VPC à l'aide d'une [AWS::EC2::RouteTable](#) ressource. La table de routage gère les itinéraires pour les instances au sein du VPC. Un [AWS::EC2::EgressOnlyInternetGateway](#) est utilisé pour créer une passerelle Internet de sortie uniquement afin de permettre la IPv6 communication pour le trafic sortant depuis les instances du VPC, tout en empêchant le trafic entrant. Une [AWS::EC2::Route](#) ressource est spécifiée pour créer un IPv6 itinéraire dans la table de routage qui dirige tout le IPv6 trafic sortant (`:::/0`) vers la passerelle Internet de sortie uniquement.

Pour plus d'informations sur les passerelles Internet de sortie uniquement, consultez [Activer le IPv6 trafic sortant à l'aide d'une passerelle Internet de sortie uniquement dans le guide de l'utilisateur Amazon VPC](#).

JSON

```
"DefaultIpv6Route": {
  "Type": "AWS::EC2::Route",
  "Properties": {
    "DestinationIpv6CidrBlock": ":::/0",
    "EgressOnlyInternetGatewayId": {
      "Ref": "EgressOnlyInternetGateway"
    },
    "RouteTableId": {
      "Ref": "RouteTable"
    }
  }
},
"EgressOnlyInternetGateway": {
  "Type": "AWS::EC2::EgressOnlyInternetGateway",
  "Properties": {
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
```



```

        "VpcId": {
            "Ref": "VPC"
        }
    },
    "VPC": {
        "Type": "AWS::EC2::VPC",
        "Properties": {
            "CidrBlock": "10.0.0.0/16"
        }
    }
}

```

YAML

```

DefaultIpv6Route:
  Type: AWS::EC2::Route
  Properties:
    DestinationIpv6CidrBlock: "::/0"
    EgressOnlyInternetGatewayId:
      Ref: "EgressOnlyInternetGateway"
    RouteTableId:
      Ref: "RouteTable"
EgressOnlyInternetGateway:
  Type: AWS::EC2::EgressOnlyInternetGateway
  Properties:
    VpcId:
      Ref: "VPC"
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: "VPC"
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: "10.0.0.0/16"

```

Extraits de modèle d'interface réseau Elastic (ENI)

Création d'une instance Amazon EC2 avec des interfaces réseau élastiques associées () ENIs

L'extrait d'exemple suivant crée une instance Amazon EC2 à l'aide d'une ressource

[AWS::EC2::Instance](#) dans le VPC et le sous-réseau Amazon spécifiés. Il associe deux interfaces

réseau (ENIs) à l'instance, associe des adresses IP élastiques aux instances par le biais de l'interface attachée ENIs et configure le groupe de sécurité pour l'accès SSH et HTTP. Les données utilisateur sont fournies à l'instance dans le cadre de la configuration de lancement lors de la création de l'instance. Les données utilisateur incluent un script codé dans un format base64 afin de garantir leur transmission à l'instance. Lorsque l'instance est lancée, le script s'exécute automatiquement dans le cadre du processus de démarrage. Il installe `ec2-net-utils`, configure les interfaces réseau et démarre le service HTTP.

Pour déterminer la Amazon Machine Image (AMI) appropriée en fonction de la région sélectionnée, l'extrait utilise une fonction `Fn::FindInMap` qui recherche des valeurs dans un mappage `RegionMap`. Ce mappage doit être défini dans le modèle plus large. Les deux interfaces réseau sont créées à l'aide de ressources [AWS::EC2::NetworkInterface](#). Les adresses IP élastiques sont spécifiées à l'aide des ressources [AWS::EC2::EIP](#) allouées au domaine `vpc`. Ces adresses IP élastiques sont associées aux interfaces réseau utilisant des ressources [AWS::EC2::EIPAssociation](#).

La section `Outputs` définit les valeurs ou les ressources auxquelles vous souhaitez accéder une fois que la pile est créée. Dans cet extrait, la sortie définie est `InstancePublicIp`, qui représente l'adresse IP publique de l'instance EC2 créée par la pile. Vous pouvez récupérer cette sortie dans l'onglet `Sortie` de la CloudFormation console ou à l'aide de la commande [describe-stacks](#).

Pour plus d'informations sur les interfaces réseau Elastic, consultez [Interfaces réseau Elastic](#).

JSON

```
"Resources": {
  "ControlPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
      "Domain": "vpc"
    }
  },
  "AssociateControlPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
      "AllocationId": {
        "Fn::GetAtt": [
          "ControlPortAddress",
          "AllocationId"
        ]
      },
      "NetworkInterfaceId": {
```

```

        "Ref": "controlXface"
      }
    }
  },
  "WebPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
      "Domain": "vpc"
    }
  },
  "AssociateWebPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
      "AllocationId": {
        "Fn::GetAtt": [
          "WebPortAddress",
          "AllocationId"
        ]
      },
      "NetworkInterfaceId": {
        "Ref": "webXface"
      }
    }
  },
  "SSHSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "VpcId": {
        "Ref": "VpcId"
      },
      "GroupDescription": "Enable SSH access via port 22",
      "SecurityGroupIngress": [
        {
          "CidrIp": "0.0.0.0/0",
          "FromPort": 22,
          "IpProtocol": "tcp",
          "ToPort": 22
        }
      ]
    }
  },
  "WebSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {

```

```
        "VpcId": {
            "Ref": "VpcId"
        },
        "GroupDescription": "Enable HTTP access via user-defined port",
        "SecurityGroupIngress": [
            {
                "CidrIp": "0.0.0.0/0",
                "FromPort": 80,
                "IpProtocol": "tcp",
                "ToPort": 80
            }
        ]
    },
    "controlXface": {
        "Type": "AWS::EC2::NetworkInterface",
        "Properties": {
            "SubnetId": {
                "Ref": "SubnetId"
            },
            "Description": "Interface for controlling traffic such as SSH",
            "GroupSet": [
                {
                    "Fn::GetAtt": [
                        "SSHSecurityGroup",
                        "GroupId"
                    ]
                }
            ],
            "SourceDestCheck": true,
            "Tags": [
                {
                    "Key": "Network",
                    "Value": "Control"
                }
            ]
        }
    },
    "webXface": {
        "Type": "AWS::EC2::NetworkInterface",
        "Properties": {
            "SubnetId": {
                "Ref": "SubnetId"
            },
        },
    },
```

```

        "Description": "Interface for web traffic",
        "GroupSet": [
            {
                "Fn::GetAtt": [
                    "WebSecurityGroup",
                    "GroupId"
                ]
            }
        ],
        "SourceDestCheck": true,
        "Tags": [
            {
                "Key": "Network",
                "Value": "Web"
            }
        ]
    }
},
"Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "ImageId": {
            "Fn::FindInMap": [
                "RegionMap",
                {
                    "Ref": "AWS::Region"
                }
            ],
            "AMI"
        ]
    },
    "KeyName": {
        "Ref": "KeyName"
    },
    "NetworkInterfaces": [
        {
            "NetworkInterfaceId": {
                "Ref": "controlXface"
            },
            "DeviceIndex": "0"
        },
        {
            "NetworkInterfaceId": {
                "Ref": "webXface"
            }
        }
    ]
}

```

```

        "DeviceIndex": "1"
      }
    ],
    "Tags": [
      {
        "Key": "Role",
        "Value": "Test Instance"
      }
    ],
    "UserData": {
      "Fn::Base64": {
        "Fn::Sub": "#!/bin/bash -xe\nyum install ec2-net-utils -y\nec2ifup
eth1\nservice httpd start\n"
      }
    }
  }
},
"Outputs": {
  "InstancePublicIp": {
    "Description": "Public IP Address of the EC2 Instance",
    "Value": {
      "Fn::GetAtt": [
        "Ec2Instance",
        "PublicIp"
      ]
    }
  }
}
}

```

YAML

```

Resources:
  ControlPortAddress:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc
  AssociateControlPort:
    Type: AWS::EC2::EIPAssociation
    Properties:
      AllocationId:
        Fn::GetAtt:
          - ControlPortAddress

```

```
- AllocationId
NetworkInterfaceId:
  Ref: controlXface
WebPortAddress:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc
AssociateWebPort:
  Type: AWS::EC2::EIPAssociation
  Properties:
    AllocationId:
      Fn::GetAtt:
        - WebPortAddress
        - AllocationId
    NetworkInterfaceId:
      Ref: webXface
SSHSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 22
        IpProtocol: tcp
        ToPort: 22
WebSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable HTTP access via user-defined port
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 80
        IpProtocol: tcp
        ToPort: 80
controlXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
    Description: Interface for controlling traffic such as SSH
```

```
GroupSet:
  - Fn::GetAtt:
    - SSHSecurityGroup
    - GroupId
SourceDestCheck: true
Tags:
  - Key: Network
    Value: Control
webXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
    Description: Interface for web traffic
    GroupSet:
      - Fn::GetAtt:
        - WebSecurityGroup
        - GroupId
    SourceDestCheck: true
    Tags:
      - Key: Network
        Value: Web
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Fn::FindInMap:
        - RegionMap
        - Ref: AWS::Region
        - AMI
    KeyName:
      Ref: KeyName
    NetworkInterfaces:
      - NetworkInterfaceId:
          Ref: controlXface
          DeviceIndex: "0"
      - NetworkInterfaceId:
          Ref: webXface
          DeviceIndex: "1"
    Tags:
      - Key: Role
        Value: Test Instance
    UserData:
      Fn::Base64: !Sub |
```



```
#!/bin/bash -xe
yum install ec2-net-utils -y
ec2ifup eth1
service httpd start
```

Outputs:**InstancePublicIp:**

Description: Public IP Address of the EC2 Instance

Value:

Fn::GetAtt:

- Ec2Instance
- PublicIp

Modèles d'exemple Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) est un service de gestion de conteneurs, qui permet d'exécuter, d'arrêter et de gérer facilement des conteneurs Docker dans un cluster d'instances Amazon Elastic Compute Cloud (Amazon EC2).

Créez un cluster avec AL2023 Amazon ECS-Optimized-AMI

Définissez un cluster qui utilise un fournisseur de capacité qui lance des instances AL2023 sur Amazon EC2.

** Important**

Pour obtenir la dernière AMI IDs, consultez l'[AMI optimisée pour Amazon ECS dans le manuel](#) Amazon Elastic Container Service Developer Guide.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.",
  "Parameters": {
    "InstanceType": {
      "Type": "String",
      "Description": "EC2 instance type",
      "Default": "t2.medium",
      "AllowedValues": [
```

```

        "t1.micro",
        "t2.2xlarge",
        "t2.large",
        "t2.medium",
        "t2.micro",
        "t2.nano",
        "t2.small",
        "t2.xlarge",
        "t3.2xlarge",
        "t3.large",
        "t3.medium",
        "t3.micro",
        "t3.nano",
        "t3.small",
        "t3.xlarge"
    ]
},
"DesiredCapacity": {
    "Type": "Number",
    "Default": "0",
    "Description": "Number of EC2 instances to launch in your ECS cluster."
},
"MaxSize": {
    "Type": "Number",
    "Default": "100",
    "Description": "Maximum number of EC2 instances that can be launched in your
ECS cluster."
},
"ECSAMI": {
    "Description": "The Amazon Machine Image ID used for the cluster",
    "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
    "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/
image_id"
},
"VpcId": {
    "Type": "AWS::EC2::VPC::Id",
    "Description": "VPC ID where the ECS cluster is launched",
    "Default": "vpc-1234567890abcdef0"
},
"SubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of subnet IDs where the EC2 instances will be launched",
    "Default": "subnet-021345abcdef67890"
}

```

```
},
"Resources": {
  "ECSCluster": {
    "Type": "AWS::ECS::Cluster",
    "Properties": {
      "ClusterSettings": [
        {
          "Name": "containerInsights",
          "Value": "enabled"
        }
      ]
    }
  },
  "ECSAutoScalingGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "DependsOn": [
      "ECSCluster",
      "EC2Role"
    ],
    "Properties": {
      "VPCZoneIdentifier": {
        "Ref": "SubnetIds"
      },
      "LaunchTemplate": {
        "LaunchTemplateId": {
          "Ref": "ContainerInstances"
        },
        "Version": {
          "Fn::GetAtt": [
            "ContainerInstances",
            "LatestVersionNumber"
          ]
        }
      },
      "MinSize": 0,
      "MaxSize": {
        "Ref": "MaxSize"
      },
      "DesiredCapacity": {
        "Ref": "DesiredCapacity"
      },
      "NewInstancesProtectedFromScaleIn": true
    },
    "UpdatePolicy": {
```

```

        "AutoScalingReplacingUpdate": {
            "WillReplace": "true"
        }
    },
    "ContainerInstances": {
        "Type": "AWS::EC2::LaunchTemplate",
        "Properties": {
            "LaunchTemplateName": "asg-launch-template",
            "LaunchTemplateData": {
                "ImageId": {
                    "Ref": "ECSAMI"
                },
                "InstanceType": {
                    "Ref": "InstanceType"
                },
                "IamInstanceProfile": {
                    "Name": {
                        "Ref": "EC2InstanceProfile"
                    }
                },
                "SecurityGroupIds": [
                    {
                        "Ref": "ContainerHostSecurityGroup"
                    }
                ],
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Sub": "
#!/bin/bash -xe\n
echo ECS_CLUSTER=${ECSCluster}
>> /etc/ecs/ecs.config\n
yum install -y aws-cfn-bootstrap\n
/opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource ContainerInstances --configsets full_install --region ${AWS::Region} &\n"
                    }
                },
                "MetadataOptions": {
                    "HttpEndpoint": "enabled",
                    "HttpTokens": "required"
                }
            }
        }
    },
    "EC2InstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {

```

```

        "Path": "/",
        "Roles": [
            {
                "Ref": "EC2Role"
            }
        ]
    },
    "CapacityProvider": {
        "Type": "AWS::ECS::CapacityProvider",
        "Properties": {
            "AutoScalingGroupProvider": {
                "AutoScalingGroupArn": {
                    "Ref": "ECSAutoScalingGroup"
                },
            },
            "ManagedScaling": {
                "InstanceWarmupPeriod": 60,
                "MinimumScalingStepSize": 1,
                "MaximumScalingStepSize": 100,
                "Status": "ENABLED",
                "TargetCapacity": 100
            },
            "ManagedTerminationProtection": "ENABLED"
        }
    },
    "CapacityProviderAssociation": {
        "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
        "Properties": {
            "CapacityProviders": [
                {
                    "Ref": "CapacityProvider"
                }
            ],
            "Cluster": {
                "Ref": "ECSCluster"
            },
            "DefaultCapacityProviderStrategy": [
                {
                    "Base": 0,
                    "CapacityProvider": {
                        "Ref": "CapacityProvider"
                    },
                    "Weight": 1
                }
            ]
        }
    }
}

```

```

        }
    ]
}
},
"ContainerHostSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the EC2 hosts that run containers",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"EC2Role": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "ec2.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        },
        "Path": "/",
        "ManagedPolicyArns": [
            "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role",
            "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
        ]
    }
},
"ECSTaskExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Statement": [

```

```

        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ecs-tasks.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": {
                        "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
                    }
                },
                "StringEquals": {
                    "aws:SourceAccount": {
                        "Fn::Sub": "${AWS::AccountId}"
                    }
                }
            }
        }
    ],
    "Path": "/",
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
    ]
},
"Outputs": {
    "ClusterName": {
        "Description": "The ECS cluster into which to launch resources",
        "Value": "ECSCluster"
    },
    "ECSTaskExecutionRole": {
        "Description": "The role used to start up a task",
        "Value": "ECSTaskExecutionRole"
    },
    "CapacityProvider": {

```

```
        "Description": "The cluster capacity provider that the service should use to
        request capacity when it wants to start up a task",
        "Value": "CapacityProvider"
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet.
  An ECS capacity provider automatically launches more EC2 instances as required
  on the fly when you request ECS to launch services or standalone tasks.
Parameters:
  InstanceType:
    Type: String
    Description: EC2 instance type
    Default: "t2.medium"
    AllowedValues:
      - t1.micro
      - t2.2xlarge
      - t2.large
      - t2.medium
      - t2.micro
      - t2.nano
      - t2.small
      - t2.xlarge
      - t3.2xlarge
      - t3.large
      - t3.medium
      - t3.micro
      - t3.nano
      - t3.small
      - t3.xlarge
  DesiredCapacity:
    Type: Number
    Default: "0"
    Description: Number of EC2 instances to launch in your ECS cluster.
  MaxSize:
    Type: Number
    Default: "100"
```


Description: Maximum number of EC2 instances that can be launched in your ECS cluster.

ECSAMI:

Description: The Amazon Machine Image ID used for the cluster

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id

VpcId:

Type: AWS::EC2::VPC::Id

Description: VPC ID where the ECS cluster is launched

Default: vpc-1234567890abcdef0

SubnetIds:

Type: List<AWS::EC2::Subnet::Id>

Description: List of subnet IDs where the EC2 instances will be launched

Default: "subnet-021345abcdef67890"

Resources:

```
# This is authorizes ECS to manage resources on your
# account on your behalf. This role is likely already created on your account
# ECSRole:
#   Type: AWS::IAM::ServiceLinkedRole
#   Properties:
#     AWSServiceName: 'ecs.amazonaws.com'
```

ECS Resources

ECSCluster:

Type: AWS::ECS::Cluster

Properties:

ClusterSettings:

- Name: containerInsights
- Value: enabled

```
# Autoscaling group. This launches the actual EC2 instances that will register
# themselves as members of the cluster, and run the docker containers.
```

ECSAutoScalingGroup:

Type: AWS::AutoScaling::AutoScalingGroup

DependsOn:

```
# This is to ensure that the ASG gets deleted first before these
# resources, when it comes to stack teardown.
```

- ECSCluster
- EC2Role

Properties:

VPCZoneIdentifier:

Ref: SubnetIds

LaunchTemplate:

LaunchTemplateId: !Ref ContainerInstances

```

    Version: !GetAtt ContainerInstances.LatestVersionNumber
    MinSize: 0
    MaxSize:
      Ref: MaxSize
    DesiredCapacity:
      Ref: DesiredCapacity
    NewInstancesProtectedFromScaleIn: true
  UpdatePolicy:
    AutoScalingReplacingUpdate:
      WillReplace: "true"
# The config for each instance that is added to the cluster
ContainerInstances:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: "asg-launch-template"
    LaunchTemplateData:
      ImageId:
        Ref: ECSAMI
      InstanceType:
        Ref: InstanceType
      IamInstanceProfile:
        Name: !Ref EC2InstanceProfile
      SecurityGroupIds:
        - !Ref ContainerHostSecurityGroup
      # This injected configuration file is how the EC2 instance
      # knows which ECS cluster on your AWS account it should be joining
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
ContainerInstances --configsets full_install --region ${AWS::Region} &
          # Disable IMDSv1, and require IMDSv2
      MetadataOptions:
        HttpEndpoint: enabled
        HttpTokens: required
    EC2InstanceProfile:
      Type: AWS::IAM::InstanceProfile
      Properties:
        Path: /
        Roles:
          - !Ref EC2Role
# Create an ECS capacity provider to attach the ASG to the ECS cluster

```

```

# so that it autoscales as we launch more containers
CapacityProvider:
  Type: AWS::ECS::CapacityProvider
  Properties:
    AutoScalingGroupProvider:
      AutoScalingGroupArn: !Ref ECSAutoScalingGroup
      ManagedScaling:
        InstanceWarmupPeriod: 60
        MinimumScalingStepSize: 1
        MaximumScalingStepSize: 100
        Status: ENABLED
        # Percentage of cluster reservation to try to maintain
        TargetCapacity: 100
      ManagedTerminationProtection: ENABLED
    # Create a cluster capacity provider association so that the cluster
    # will use the capacity provider
  CapacityProviderAssociation:
    Type: AWS::ECS::ClusterCapacityProviderAssociations
    Properties:
      CapacityProviders:
        - !Ref CapacityProvider
      Cluster: !Ref ECSCluster
      DefaultCapacityProviderStrategy:
        - Base: 0
          CapacityProvider: !Ref CapacityProvider
          Weight: 1
    # A security group for the EC2 hosts that will run the containers.
    # This can be used to limit incoming traffic to or outgoing traffic
    # from the container's host EC2 instance.
  ContainerHostSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Access to the EC2 hosts that run containers
      VpcId:
        Ref: VpcId
    # Role for the EC2 hosts. This allows the ECS agent on the EC2 hosts
    # to communicate with the ECS control plane, as well as download the docker
    # images from ECR to run on your host.
  EC2Role:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow

```

```

    Principal:
      Service:
        - ec2.amazonaws.com
    Action:
      - sts:AssumeRole
    Path: /
    ManagedPolicyArns:
      # See reference: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/
security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerServiceforEC2Role
        - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      # This managed policy allows us to connect to the instance using SSM
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
    # This is a role which is used within Fargate to allow the Fargate agent
    # to download images, and upload logs.
    ECSTaskExecutionRole:
      Type: AWS::IAM::Role
      Properties:
        AssumeRolePolicyDocument:
          Statement:
            - Effect: Allow
              Principal:
                Service:
                  - ecs-tasks.amazonaws.com
              Action:
                - sts:AssumeRole
          Condition:
            ArnLike:
              aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
            StringEquals:
              aws:SourceAccount: !Sub ${AWS::AccountId}
        Path: /
        # This role enables all features of ECS. See reference:
        # https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-iam-
awsmanpol.html#security-iam-awsmanpol-AmazonECSTaskExecutionRolePolicy
        ManagedPolicyArns:
          - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
    Outputs:
      ClusterName:
        Description: The ECS cluster into which to launch resources
        Value: ECSCluster
      ECSTaskExecutionRole:
        Description: The role used to start up a task
        Value: ECSTaskExecutionRole

```

CapacityProvider:

Description: The cluster capacity provider that the service should use to request capacity when it wants to start up a task

Value: CapacityProvider

Déployer un service

Le modèle suivant définit un service qui utilise le fournisseur de capacité pour demander AL2023 la capacité à exécuter. Les conteneurs seront lancés sur les AL2023 instances au fur et à mesure de leur mise en ligne :

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.",
  "Parameters": {
    "VpcId": {
      "Type": "String",
      "Description": "The VPC that the service is running inside of"
    },
    "PublicSubnetIds": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "List of public subnet ID's to put the load balancer in"
    },
    "PrivateSubnetIds": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "List of private subnet ID's that the AWS VPC tasks are in"
    },
    "ClusterName": {
      "Type": "String",
      "Description": "The name of the ECS cluster into which to launch capacity."
    },
    "ECSTaskExecutionRole": {
      "Type": "String",
      "Description": "The role used to start up an ECS task"
    },
    "CapacityProvider": {
      "Type": "String",
```

```
    "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task"
  },
  "ServiceName": {
    "Type": "String",
    "Default": "web",
    "Description": "A name for the service"
  },
  "ImageUrl": {
    "Type": "String",
    "Default": "public.ecr.aws/docker/library/nginx:latest",
    "Description": "The url of a docker image that contains the application
process that will handle the traffic for this service"
  },
  "ContainerCpu": {
    "Type": "Number",
    "Default": 256,
    "Description": "How much CPU to give the container. 1024 is 1 CPU"
  },
  "ContainerMemory": {
    "Type": "Number",
    "Default": 512,
    "Description": "How much memory in megabytes to give the container"
  },
  "ContainerPort": {
    "Type": "Number",
    "Default": 80,
    "Description": "What port that the application expects traffic on"
  },
  "DesiredCount": {
    "Type": "Number",
    "Default": 2,
    "Description": "How many copies of the service task to run"
  }
},
"Resources": {
  "TaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "Family": {
        "Ref": "ServiceName"
      },
      "Cpu": {
        "Ref": "ContainerCpu"
      }
    }
  }
}
```

```
    },
    "Memory": {
      "Ref": "ContainerMemory"
    },
    "NetworkMode": "awsvpc",
    "RequiresCompatibilities": [
      "EC2"
    ],
    "ExecutionRoleArn": {
      "Ref": "ECSTaskExecutionRole"
    },
    "ContainerDefinitions": [
      {
        "Name": {
          "Ref": "ServiceName"
        },
        "Cpu": {
          "Ref": "ContainerCpu"
        },
        "Memory": {
          "Ref": "ContainerMemory"
        },
        "Image": {
          "Ref": "ImageUrl"
        },
        "PortMappings": [
          {
            "ContainerPort": {
              "Ref": "ContainerPort"
            },
            "HostPort": {
              "Ref": "ContainerPort"
            }
          }
        ],
        "LogConfiguration": {
          "LogDriver": "awslogs",
          "Options": {
            "mode": "non-blocking",
            "max-buffer-size": "25m",
            "awslogs-group": {
              "Ref": "LogGroup"
            },
            "awslogs-region": {
```

```

        "Ref": "AWS::Region"
      },
      "awslogs-stream-prefix": {
        "Ref": "ServiceName"
      }
    }
  }
}
],
}
},
"Service": {
  "Type": "AWS::ECS::Service",
  "DependsOn": "PublicLoadBalancerListener",
  "Properties": {
    "ServiceName": {
      "Ref": "ServiceName"
    },
    "Cluster": {
      "Ref": "ClusterName"
    },
    "PlacementStrategies": [
      {
        "Field": "attribute:ecs.availability-zone",
        "Type": "spread"
      },
      {
        "Field": "cpu",
        "Type": "binpack"
      }
    ],
    "CapacityProviderStrategy": [
      {
        "Base": 0,
        "CapacityProvider": {
          "Ref": "CapacityProvider"
        },
        "Weight": 1
      }
    ],
    "NetworkConfiguration": {
      "AwsvpcConfiguration": {
        "SecurityGroups": [
          {

```



```

        "Ref": "ServiceSecurityGroup"
      }
    ],
    "Subnets": {
      "Ref": "PrivateSubnetIds"
    }
  },
  "DeploymentConfiguration": {
    "MaximumPercent": 200,
    "MinimumHealthyPercent": 75
  },
  "DesiredCount": {
    "Ref": "DesiredCount"
  },
  "TaskDefinition": {
    "Ref": "TaskDefinition"
  },
  "LoadBalancers": [
    {
      "ContainerName": {
        "Ref": "ServiceName"
      },
      "ContainerPort": {
        "Ref": "ContainerPort"
      },
      "TargetGroupArn": {
        "Ref": "ServiceTargetGroup"
      }
    }
  ]
},
"ServiceSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Security group for service",
    "VpcId": {
      "Ref": "VpcId"
    }
  }
},
"ServiceTargetGroup": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",

```

```

    "Properties": {
      "HealthCheckIntervalSeconds": 6,
      "HealthCheckPath": "/",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 2,
      "TargetType": "ip",
      "Port": {
        "Ref": "ContainerPort"
      },
      "Protocol": "HTTP",
      "UnhealthyThresholdCount": 10,
      "VpcId": {
        "Ref": "VpcId"
      },
      "TargetGroupAttributes": [
        {
          "Key": "deregistration_delay.timeout_seconds",
          "Value": 0
        }
      ]
    }
  },
  "PublicLoadBalancerSG": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Access to the public facing load balancer",
      "VpcId": {
        "Ref": "VpcId"
      },
      "SecurityGroupIngress": [
        {
          "CidrIp": "0.0.0.0/0",
          "IpProtocol": -1
        }
      ]
    }
  },
  "PublicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internet-facing",
      "LoadBalancerAttributes": [
        {

```

```

        "Key": "idle_timeout.timeout_seconds",
        "Value": "30"
      }
    ],
    "Subnets": {
      "Ref": "PublicSubnetIds"
    },
    "SecurityGroups": [
      {
        "Ref": "PublicLoadBalancerSG"
      }
    ]
  }
},
"PublicLoadBalancerListener": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [
      {
        "Type": "forward",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": {
                "Ref": "ServiceTargetGroup"
              },
              "Weight": 100
            }
          ]
        }
      }
    ]
  },
  "LoadBalancerArn": {
    "Ref": "PublicLoadBalancer"
  },
  "Port": 80,
  "Protocol": "HTTP"
},
"ServiceIngressfromLoadBalancer": {
  "Type": "AWS::EC2::SecurityGroupIngress",
  "Properties": {
    "Description": "Ingress from the public ALB",
    "GroupId": {

```

```

        "Ref": "ServiceSecurityGroup"
      },
      "IpProtocol": -1,
      "SourceSecurityGroupId": {
        "Ref": "PublicLoadBalancerSG"
      }
    }
  },
  "LogGroup": {
    "Type": "AWS::Logs::LogGroup"
  }
}
}

```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: >-

An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.

Parameters:

VpcId:

Type: String

Description: The VPC that the service is running inside of

PublicSubnetIds:

Type: 'List<AWS::EC2::Subnet::Id>'

Description: List of public subnet ID's to put the load balancer in

PrivateSubnetIds:

Type: 'List<AWS::EC2::Subnet::Id>'

Description: List of private subnet ID's that the AWS VPC tasks are in

ClusterName:

Type: String

Description: The name of the ECS cluster into which to launch capacity.

ECSTaskExecutionRole:

Type: String

Description: The role used to start up an ECS task

CapacityProvider:

Type: String

Description: >-

The cluster capacity provider that the service should use to request capacity when it wants to start up a task

```
ServiceName:
  Type: String
  Default: web
  Description: A name for the service
ImageUrl:
  Type: String
  Default: 'public.ecr.aws/docker/library/nginx:latest'
  Description: >-
    The url of a docker image that contains the application process that will
    handle the traffic for this service
ContainerCpu:
  Type: Number
  Default: 256
  Description: How much CPU to give the container. 1024 is 1 CPU
ContainerMemory:
  Type: Number
  Default: 512
  Description: How much memory in megabytes to give the container
ContainerPort:
  Type: Number
  Default: 80
  Description: What port that the application expects traffic on
DesiredCount:
  Type: Number
  Default: 2
  Description: How many copies of the service task to run
Resources:
  TaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      Family: !Ref ServiceName
      Cpu: !Ref ContainerCpu
      Memory: !Ref ContainerMemory
      NetworkMode: awsvpc
      RequiresCompatibilities:
        - EC2
      ExecutionRoleArn: !Ref ECSTaskExecutionRole
      ContainerDefinitions:
        - Name: !Ref ServiceName
          Cpu: !Ref ContainerCpu
          Memory: !Ref ContainerMemory
          Image: !Ref ImageUrl
          PortMappings:
            - ContainerPort: !Ref ContainerPort
```

```
    HostPort: !Ref ContainerPort
  LogConfiguration:
    LogDriver: awslogs
    Options:
      mode: non-blocking
      max-buffer-size: 25m
      awslogs-group: !Ref LogGroup
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Ref ServiceName
Service:
  Type: AWS::ECS::Service
  DependsOn: PublicLoadBalancerListener
  Properties:
    ServiceName: !Ref ServiceName
    Cluster: !Ref ClusterName
    PlacementStrategies:
      - Field: 'attribute:ecs.availability-zone'
        Type: spread
      - Field: cpu
        Type: binpack
    CapacityProviderStrategy:
      - Base: 0
        CapacityProvider: !Ref CapacityProvider
        Weight: 1
    NetworkConfiguration:
      AwsvpcConfiguration:
        SecurityGroups:
          - !Ref ServiceSecurityGroup
        Subnets: !Ref PrivateSubnetIds
    DeploymentConfiguration:
      MaximumPercent: 200
      MinimumHealthyPercent: 75
    DesiredCount: !Ref DesiredCount
    TaskDefinition: !Ref TaskDefinition
    LoadBalancers:
      - ContainerName: !Ref ServiceName
        ContainerPort: !Ref ContainerPort
        TargetGroupArn: !Ref ServiceTargetGroup
ServiceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for service
    VpcId: !Ref VpcId
ServiceTargetGroup:
```

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 6
HealthCheckPath: /
HealthCheckProtocol: HTTP
HealthCheckTimeoutSeconds: 5
HealthyThresholdCount: 2
TargetType: ip
Port: !Ref ContainerPort
Protocol: HTTP
UnhealthyThresholdCount: 10
VpcId: !Ref VpcId
TargetGroupAttributes:

- Key: deregistration_delay.timeout_seconds
Value: 0

PublicLoadBalancerSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Access to the public facing load balancer
VpcId: !Ref VpcId
SecurityGroupIngress:

- CidrIp: 0.0.0.0/0
IpProtocol: -1

PublicLoadBalancer:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Scheme: internet-facing
LoadBalancerAttributes:

- Key: idle_timeout.timeout_seconds
Value: '30'

Subnets: !Ref PublicSubnetIds
SecurityGroups:

- !Ref PublicLoadBalancerSG

PublicLoadBalancerListener:

Type: AWS::ElasticLoadBalancingV2::Listener

Properties:

DefaultActions:

- Type: forward
ForwardConfig:
TargetGroups:
 - TargetGroupArn: !Ref ServiceTargetGroup
Weight: 100

LoadBalancerArn: !Ref PublicLoadBalancer
Port: 80

```
Protocol: HTTP
ServiceIngressfromLoadBalancer:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    Description: Ingress from the public ALB
    GroupId: !Ref ServiceSecurityGroup
    IpProtocol: -1
    SourceSecurityGroupId: !Ref PublicLoadBalancerSG
LogGroup:
  Type: AWS::Logs::LogGroup
```

Exemple de modèle Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) est un service de stockage de fichiers pour les instances Amazon Elastic Compute Cloud (Amazon EC2). Avec Amazon EFS, vos applications disposent de l'espace de stockage dont elles en ont besoin, car la capacité de stockage augmente ou diminue automatiquement lorsque vous ajoutez ou supprimez des fichiers.

L'exemple de modèle suivant déploie EC2 des instances (dans un groupe Auto Scaling) associées à un système de fichiers Amazon EFS. Pour associer les instances au système de fichiers, celles-ci exécutent le script d'assistant cfn-init, qui télécharge et installe le package yum `nfs-utils`, qui crée un répertoire, puis qui utilise le nom DNS du système de fichiers pour monter le système de fichiers sans ce répertoire. Le nom DNS du système de fichiers correspond à l'adresse IP d'une cible de montage dans la zone de disponibilité de l' EC2instance Amazon. Pour plus d'informations sur la structure du nom DNS, consultez [Montage des systèmes de fichiers](#) dans le Amazon Elastic File System Guide de l'utilisateur.

Pour mesurer l'activité du système de fichiers réseau, le modèle inclut des CloudWatch métriques Amazon personnalisées. Le modèle crée également un VPC, un sous-réseau et des groupes de sécurité. Pour permettre aux instances de communiquer avec le système de fichiers, le DNS doit être activé sur le VPC, et la cible de montage et les EC2 instances doivent se trouver dans la même zone de disponibilité (AZ), spécifiée par le sous-réseau.

Le groupe de sécurité de la cible de montage permet une connexion réseau au port TCP 2049, qui est nécessaire pour qu'un NFSv4 client puisse monter un système de fichiers. Pour plus d'informations sur les groupes de sécurité pour les EC2 instances et les cibles de montage, consultez [Security](#) in the [Amazon Elastic File System User Guide](#).

 Note

Si vous effectuez une mise à jour de la cible de montage qui entraîne son remplacement, les instances ou les applications qui utilisent le système de fichiers associé peuvent être interrompues. Ceci peut entraîner la perte des écritures non enregistrées. Pour éviter les perturbations, arrêtez vos instances lorsque vous mettez à jour la cible de montage en définissant la capacité souhaitée sur zéro. Cette procédure permet aux instances de démonter le système de fichiers avant la suppression de la cible de montage. Une fois que la mise à jour de montage est terminée, démarrez vos instances lors d'une mise à jour ultérieure en définissant la capacité souhaitée.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. **WARNING** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.",
  "Parameters": {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [
        "t1.micro",
        "t2.nano",
        "t2.micro",
        "t2.small",
        "t2.medium",
        "t2.large",
        "m1.small",
        "m1.medium",
        "m1.large",
        "m1.xlarge",
        "m2.xlarge",
        "m2.2xlarge",
        "m2.4xlarge",
        "m3.medium",
        "m3.large",
```

```
    "m3.xlarge",
    "m3.2xlarge",
    "m4.large",
    "m4.xlarge",
    "m4.2xlarge",
    "m4.4xlarge",
    "m4.10xlarge",
    "c1.medium",
    "c1.xlarge",
    "c3.large",
    "c3.xlarge",
    "c3.2xlarge",
    "c3.4xlarge",
    "c3.8xlarge",
    "c4.large",
    "c4.xlarge",
    "c4.2xlarge",
    "c4.4xlarge",
    "c4.8xlarge",
    "g2.2xlarge",
    "g2.8xlarge",
    "r3.large",
    "r3.xlarge",
    "r3.2xlarge",
    "r3.4xlarge",
    "r3.8xlarge",
    "i2.xlarge",
    "i2.2xlarge",
    "i2.4xlarge",
    "i2.8xlarge",
    "d2.xlarge",
    "d2.2xlarge",
    "d2.4xlarge",
    "d2.8xlarge",
    "hi1.4xlarge",
    "hs1.8xlarge",
    "cr1.8xlarge",
    "cc2.8xlarge",
    "cg1.4xlarge"
  ],
  "ConstraintDescription" : "must be a valid EC2 instance type."
},
"KeyName": {
  "Type": "AWS::EC2::KeyPair::KeyName",
```

```

    "Description": "Name of an existing EC2 key pair to enable SSH access to the EC2
instances"
  },
  "AsgMaxSize": {
    "Type": "Number",
    "Description": "Maximum size and initial desired capacity of Auto Scaling Group",
    "Default": "2"
  },
  "SSHLocation" : {
    "Description" : "The IP address range that can be used to connect to the EC2
instances by using SSH",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "VolumeName" : {
    "Description" : "The name to be used for the EFS volume",
    "Type": "String",
    "MinLength": "1",
    "Default": "myEFSvolume"
  },
  "MountPoint" : {
    "Description" : "The Linux mount point for the EFS volume",
    "Type": "String",
    "MinLength": "1",
    "Default": "myEFSvolume"
  }
},
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"      : { "Arch" : "HVM64"  },
    "t2.nano"      : { "Arch" : "HVM64"  },
    "t2.micro"      : { "Arch" : "HVM64"  },
    "t2.small"     : { "Arch" : "HVM64"  },
    "t2.medium"    : { "Arch" : "HVM64"  },
    "t2.large"     : { "Arch" : "HVM64"  },
    "m1.small"     : { "Arch" : "HVM64"  },
    "m1.medium"    : { "Arch" : "HVM64"  },
    "m1.large"     : { "Arch" : "HVM64"  },
    "m1.xlarge"    : { "Arch" : "HVM64"  },
    "m2.xlarge"    : { "Arch" : "HVM64"  },

```

```

    "m2.2xlarge" : { "Arch" : "HVM64" },
    "m2.4xlarge" : { "Arch" : "HVM64" },
    "m3.medium"  : { "Arch" : "HVM64" },
    "m3.large"   : { "Arch" : "HVM64" },
    "m3.xlarge"  : { "Arch" : "HVM64" },
    "m3.2xlarge" : { "Arch" : "HVM64" },
    "m4.large"   : { "Arch" : "HVM64" },
    "m4.xlarge"  : { "Arch" : "HVM64" },
    "m4.2xlarge" : { "Arch" : "HVM64" },
    "m4.4xlarge" : { "Arch" : "HVM64" },
    "m4.10xlarge" : { "Arch" : "HVM64" },
    "c1.medium"  : { "Arch" : "HVM64" },
    "c1.xlarge"  : { "Arch" : "HVM64" },
    "c3.large"   : { "Arch" : "HVM64" },
    "c3.xlarge"  : { "Arch" : "HVM64" },
    "c3.2xlarge" : { "Arch" : "HVM64" },
    "c3.4xlarge" : { "Arch" : "HVM64" },
    "c3.8xlarge" : { "Arch" : "HVM64" },
    "c4.large"   : { "Arch" : "HVM64" },
    "c4.xlarge"  : { "Arch" : "HVM64" },
    "c4.2xlarge" : { "Arch" : "HVM64" },
    "c4.4xlarge" : { "Arch" : "HVM64" },
    "c4.8xlarge" : { "Arch" : "HVM64" },
    "g2.2xlarge" : { "Arch" : "HVMG2" },
    "g2.8xlarge" : { "Arch" : "HVMG2" },
    "r3.large"   : { "Arch" : "HVM64" },
    "r3.xlarge"  : { "Arch" : "HVM64" },
    "r3.2xlarge" : { "Arch" : "HVM64" },
    "r3.4xlarge" : { "Arch" : "HVM64" },
    "r3.8xlarge" : { "Arch" : "HVM64" },
    "i2.xlarge"  : { "Arch" : "HVM64" },
    "i2.2xlarge" : { "Arch" : "HVM64" },
    "i2.4xlarge" : { "Arch" : "HVM64" },
    "i2.8xlarge" : { "Arch" : "HVM64" },
    "d2.xlarge"  : { "Arch" : "HVM64" },
    "d2.2xlarge" : { "Arch" : "HVM64" },
    "d2.4xlarge" : { "Arch" : "HVM64" },
    "d2.8xlarge" : { "Arch" : "HVM64" },
    "hi1.4xlarge" : { "Arch" : "HVM64" },
    "hs1.8xlarge" : { "Arch" : "HVM64" },
    "cr1.8xlarge" : { "Arch" : "HVM64" },
    "cc2.8xlarge" : { "Arch" : "HVM64" }
  },
  "AWSRegionArch2AMI" : {

```

```

        "us-east-1"      : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
        "us-west-2"      : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
        "us-west-1"      : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
        "eu-west-1"      : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
        "eu-west-2"      : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
        "eu-west-3"      : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
        "eu-central-1"   : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},
        "ap-northeast-1" : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
        "ap-northeast-2" : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
        "ap-northeast-3" : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
        "ap-southeast-1" : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
        "ap-southeast-2" : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
        "ap-south-1"     : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},
        "us-east-2"      : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
        "ca-central-1"   : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
        "sa-east-1"      : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-north-1"     : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-northwest-1" : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
},
"Resources": {
    "CloudWatchPutMetricsRole" : {
        "Type" : "AWS::IAM::Role",
        "Properties" : {
            "AssumeRolePolicyDocument" : {
                "Statement" : [ {
                    "Effect" : "Allow",
                    "Principal" : {
                        "Service" : [ "ec2.amazonaws.com" ]

```

```

        },
        "Action" : [ "sts:AssumeRole" ]
    } ]
},
"Path" : "/"
}
},
"CloudWatchPutMetricsRolePolicy" : {
    "Type" : "AWS::IAM::Policy",
    "Properties" : {
        "PolicyName" : "CloudWatch_PutMetricData",
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "CloudWatchPutMetricData",
                    "Effect": "Allow",
                    "Action": ["cloudwatch:PutMetricData"],
                    "Resource": ["*"]
                }
            ]
        },
        "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
    }
},
"CloudWatchPutMetricsInstanceProfile" : {
    "Type" : "AWS::IAM::InstanceProfile",
    "Properties" : {
        "Path" : "/",
        "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
    }
},
"VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
        "EnableDnsSupport" : "true",
        "EnableDnsHostnames" : "true",
        "CidrBlock": "10.0.0.0/16",
        "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
    }
},
"InternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway",
    "Properties" : {

```

```

    "Tags" : [
      { "Key" : "Application", "Value" : { "Ref" : "AWS::StackName" } },
      { "Key" : "Network", "Value" : "Public" }
    ]
  },
  "GatewayToInternet" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "InternetGatewayId" : { "Ref" : "InternetGateway" }
    }
  },
  "RouteTable":{
    "Type":"AWS::EC2::RouteTable",
    "Properties":{
      "VpcId": {"Ref":"VPC"}
    }
  },
  "SubnetRouteTableAssoc": {
    "Type" : "AWS::EC2::SubnetRouteTableAssociation",
    "Properties" : {
      "RouteTableId" : {"Ref":"RouteTable"},
      "SubnetId" : {"Ref":"Subnet"}
    }
  },
  "InternetGatewayRoute": {
    "Type":"AWS::EC2::Route",
    "Properties":{
      "DestinationCidrBlock":"0.0.0.0/0",
      "RouteTableId":{"Ref":"RouteTable"},
      "GatewayId":{"Ref":"InternetGateway"}
    }
  },
  "Subnet": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": { "Ref": "VPC" },
      "CidrBlock": "10.0.0.0/24",
      "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
    }
  },
  "InstanceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",

```

```

    "Properties": {
      "VpcId": { "Ref": "VPC" },
      "GroupDescription": "Enable SSH access via port 22",
      "SecurityGroupIngress": [
        { "IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "CidrIp": { "Ref":
"SSHLocation" } },
        { "IpProtocol": "tcp", "FromPort": 80, "ToPort": 80, "CidrIp": "0.0.0.0/0" }
      ]
    },
    "MountTargetSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId": { "Ref": "VPC" },
        "GroupDescription": "Security group for mount target",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 2049,
            "ToPort": 2049,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    },
    "FileSystem": {
      "Type": "AWS::EFS::FileSystem",
      "Properties": {
        "PerformanceMode": "generalPurpose",
        "FileSystemTags": [
          {
            "Key": "Name",
            "Value": { "Ref" : "VolumeName" }
          }
        ]
      }
    },
    "MountTarget": {
      "Type": "AWS::EFS::MountTarget",
      "Properties": {
        "FileSystemId": { "Ref": "FileSystem" },
        "SubnetId": { "Ref": "Subnet" },
        "SecurityGroups": [ { "Ref": "MountTargetSecurityGroup" } ]
      }
    }
  }

```



```

},
"LaunchConfiguration": {
  "Type": "AWS::AutoScaling::LaunchConfiguration",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "configSets" : {
        "MountConfig" : [ "setup", "mount" ]
      },
      "setup" : {
        "packages" : {
          "yum" : {
            "nfs-utils" : []
          }
        },
        "files" : {
          "/home/ec2-user/post_nfsstat" : {
            "content" : { "Fn::Join" : [ "", [
              "#!/bin/bash\n",
              "\n",
              "INPUT=\"$(cat)\"\n",
              "CW_JSON_OPEN='{ \"Namespace\": \"EFS\", \"MetricData\": [ '\n",
              "CW_JSON_CLOSE=' ] }'\n",
              "CW_JSON_METRIC=''\n",
              "METRIC_COUNTER=0\n",
              "\n",
              "for COL in 1 2 3 4 5 6; do\n",
              "\n",
              "  COUNTER=0\n",
              "  METRIC_FIELD=$COL\n",
              "  DATA_FIELD=$(( $COL + ( $COL - 1 ) ) )\n",
              "\n",
              "  while read line; do\n",
              "    if [[ COUNTER -gt 0 ]]; then\n",
              "\n",
              "      LINE=`echo $line | tr -s ' ' '\n`,\n",
              "      AWS_COMMAND=\"aws cloudwatch put-metric-data --region \"",
              { "Ref": "AWS::Region" }, "\"\n",
              "      MOD=$(( $COUNTER % 2 ))\n",
              "\n",
              "      if [ $MOD -eq 1 ]; then\n",
              "        METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`\n",
              "\n",
              "      else\n",
              "        METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`\n",

```

```

        fi\n",
        "\n",
        "        if [[ -n \"\$METRIC_NAME\" && -n \"\$METRIC_VALUE\" ]]; then\n",
        "\n",
        "            INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-\n",
        "data/instance-id)\n",
        "            CW_JSON_METRIC=\"\$CW_JSON_METRIC { \\\"MetricName\\\": \\\"\n",
        "\\\"\$METRIC_NAME\\\", \\\"Dimensions\\\": [{\\\"Name\\\": \\\"InstanceId\\\", \\\"Value\\\": \\\"\n",
        "\\\"\$INSTANCE_ID\\\"} ], \\\"Value\\\": \$METRIC_VALUE },\\\"\\n",
        "            unset METRIC_NAME\n",
        "            unset METRIC_VALUE\n",
        "\n",
        "            METRIC_COUNTER=$((METRIC_COUNTER+1))\n",
        "            if [ $METRIC_COUNTER -eq 20 ]; then\n",
        "                # 20 is max metric collection size, so we have to\n",
        "submit here\n",
        "                aws cloudwatch put-metric-data --region ", { "Ref":\n",
        "\"AWS::Region\" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}\n",
        "$CW_JSON_CLOSE`\"\n",
        "\n",
        "                # reset\n",
        "                METRIC_COUNTER=0\n",
        "                CW_JSON_METRIC=''\n",
        "            fi\n",
        "        fi\n",
        "\n",
        "        COUNTER=$((COUNTER+1))\n",
        "        fi\n",
        "\n",
        "        if [[ \"\$line\" == \"Client nfs v4:\" ]]; then\n",
        "            # the next line is the good stuff\n",
        "            COUNTER=$((COUNTER+1))\n",
        "            fi\n",
        "        done <<< \"\$INPUT\"\n",
        "done\n",
        "\n",
        "# submit whatever is left\n",
        "aws cloudwatch put-metric-data --region ", { "Ref":\n",
        "\"AWS::Region\" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}\n",
        "$CW_JSON_CLOSE`\"\n",
        "    ] ] },\n",
        "mode": "000755",

```

```

        "owner": "ec2-user",
        "group": "ec2-user"
    },
    "/home/ec2-user/crontab" : {
        "content" : { "Fn::Join" : [ "", [
            "** * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
        ] ] },
        "owner": "ec2-user",
        "group": "ec2-user"
    }
},
"commands" : {
    "01_createdir" : {
        "command" : {"Fn::Join" : [ "", [ "mkdir /", { "Ref" :
"MountPoint" }]]}
    }
},
"mount" : {
    "commands" : {
        "01_mount" : {
            "command" : { "Fn::Sub": "sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}" }
        },
        "02_permissions" : {
            "command" : {"Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
{ "Ref" : "MountPoint" }]]}
        }
    }
},
"Properties": {
    "AssociatePublicIpAddress" : true,
    "ImageId": {
        "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
            "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
"Arch" ]
        } ]
    },
    "InstanceType": { "Ref": "InstanceType" },
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],

```

```

    "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
    "UserData"           : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "#!/bin/bash -xe\n",
        "yum install -y aws-cfn-bootstrap\n",

        "/opt/aws/bin/cfn-init -v ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource LaunchConfiguration ",
        "    --configsets MountConfig ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n",

        "crontab /home/ec2-user/crontab\n",

        "/opt/aws/bin/cfn-signal -e $? ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource AutoScalingGroup ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n"
    ] ] ] }
  }
},
"AutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": ["MountTarget", "GatewayToInternet"],
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT15M",
      "Count"   : { "Ref": "AsgMaxSize" }
    }
  },
  "Properties": {
    "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
    "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
    "MinSize": "1",
    "MaxSize": { "Ref": "AsgMaxSize" },
    "DesiredCapacity": { "Ref": "AsgMaxSize" },
    "Tags": [ {
      "Key": "Name",
      "Value": "EFS FileSystem Mounted Instance",
      "PropagateAtLaunch": "true"
    } ]
  }
},
"Outputs" : {

```

```
"MountTargetID" : {
  "Description" : "Mount target ID",
  "Value" : { "Ref" : "MountTarget" }
},
"FileSystemID" : {
  "Description" : "File system ID",
  "Value" : { "Ref" : "FileSystem" }
}
}
```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. ****WARNING**** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.

Parameters:

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.small

AllowedValues:

- t1.micro
- t2.nano
- t2.micro
- t2.small
- t2.medium
- t2.large
- m1.small
- m1.medium
- m1.large
- m1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- m3.medium
- m3.large
- m3.xlarge
- m3.2xlarge
- m4.large
- m4.xlarge

- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- c1.medium
- c1.xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- g2.2xlarge
- g2.8xlarge
- r3.large
- r3.xlarge
- r3.2xlarge
- r3.4xlarge
- r3.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- d2.xlarge
- d2.2xlarge
- d2.4xlarge
- d2.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- cr1.8xlarge
- cc2.8xlarge
- cg1.4xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Type: AWS::EC2::KeyPair::KeyName

Description: Name of an existing EC2 key pair to enable SSH access to the ECS instances

AsgMaxSize:

Type: Number

Description: Maximum size and initial desired capacity of Auto Scaling Group

Default: '2'

SSHLocation:

Description: The IP address range that can be used to connect to the EC2 instances by using SSH

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

VolumeName:

Description: The name to be used for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

MountPoint:

Description: The Linux mount point for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

Mappings:

AWSInstanceType2Arch:

```
t1.micro:
```

Arch: HVM64

t2.nano:

Arch: HVM64

t2.micro:

Arch: HVM64

t2.small:

Arch: HVM64

```
t2.medium:
```

Arch: HVM64

t2.large:

Arch: HVM64

```
m1.small:
```

Arch: HVM64

m1.medium:

Arch: HVM64

```
m1.large:
```

Arch: HVM64

m1.xlarge:

Arch: HVM64

m2.xlarge:

Arch: HVM64

m2.2xlarge:

```
    Arch: HVM64
m2.4xlarge:
    Arch: HVM64
m3.medium:
    Arch: HVM64
m3.large:
    Arch: HVM64
m3.xlarge:
    Arch: HVM64
m3.2xlarge:
    Arch: HVM64
m4.large:
    Arch: HVM64
m4.xlarge:
    Arch: HVM64
m4.2xlarge:
    Arch: HVM64
m4.4xlarge:
    Arch: HVM64
m4.10xlarge:
    Arch: HVM64
c1.medium:
    Arch: HVM64
c1.xlarge:
    Arch: HVM64
c3.large:
    Arch: HVM64
c3.xlarge:
    Arch: HVM64
c3.2xlarge:
    Arch: HVM64
c3.4xlarge:
    Arch: HVM64
c3.8xlarge:
    Arch: HVM64
c4.large:
    Arch: HVM64
c4.xlarge:
    Arch: HVM64
c4.2xlarge:
    Arch: HVM64
c4.4xlarge:
    Arch: HVM64
c4.8xlarge:
```



```
    Arch: HVM64
g2.2xlarge:
    Arch: HVMG2
g2.8xlarge:
    Arch: HVMG2
r3.large:
    Arch: HVM64
r3.xlarge:
    Arch: HVM64
r3.2xlarge:
    Arch: HVM64
r3.4xlarge:
    Arch: HVM64
r3.8xlarge:
    Arch: HVM64
i2.xlarge:
    Arch: HVM64
i2.2xlarge:
    Arch: HVM64
i2.4xlarge:
    Arch: HVM64
i2.8xlarge:
    Arch: HVM64
d2.xlarge:
    Arch: HVM64
d2.2xlarge:
    Arch: HVM64
d2.4xlarge:
    Arch: HVM64
d2.8xlarge:
    Arch: HVM64
hi1.4xlarge:
    Arch: HVM64
hs1.8xlarge:
    Arch: HVM64
cr1.8xlarge:
    Arch: HVM64
cc2.8xlarge:
    Arch: HVM64
AWSRegionArch2AMI:
us-east-1:
    HVM64: ami-0ff8a91507f77f867
    HVMG2: ami-0a584ac55a7631c0c
us-west-2:
```

```
HVM64: ami-a0cfeed8
HVMG2: ami-0e09505bc235aa82d
us-west-1:
  HVM64: ami-0bdb828fd58c52235
  HVMG2: ami-066ee5fd4a9ef77f1
eu-west-1:
  HVM64: ami-047bb4163c506cd98
  HVMG2: ami-0a7c483d527806435
eu-west-2:
  HVM64: ami-f976839e
  HVMG2: NOT_SUPPORTED
eu-west-3:
  HVM64: ami-0ebc281c20e89ba4b
  HVMG2: NOT_SUPPORTED
eu-central-1:
  HVM64: ami-0233214e13e500f77
  HVMG2: ami-06223d46a6d0661c7
ap-northeast-1:
  HVM64: ami-06cd52961ce9f0d85
  HVMG2: ami-053cdd503598e4a9d
ap-northeast-2:
  HVM64: ami-0a10b2721688ce9d2
  HVMG2: NOT_SUPPORTED
ap-northeast-3:
  HVM64: ami-0d98120a9fb693f07
  HVMG2: NOT_SUPPORTED
ap-southeast-1:
  HVM64: ami-08569b978cc4dfa10
  HVMG2: ami-0be9df32ae9f92309
ap-southeast-2:
  HVM64: ami-09b42976632b27e9b
  HVMG2: ami-0a9ce9fecc3d1daf8
ap-south-1:
  HVM64: ami-0912f71e06545ad88
  HVMG2: ami-097b15e89dbdcfcf4
us-east-2:
  HVM64: ami-0b59bfac6be064b78
  HVMG2: NOT_SUPPORTED
ca-central-1:
  HVM64: ami-0b18956f
  HVMG2: NOT_SUPPORTED
sa-east-1:
  HVM64: ami-07b14488da8ea02a0
  HVMG2: NOT_SUPPORTED
```

```
cn-north-1:
  HVM64: ami-0a4eaf6c4454eda75
  HVMG2: NOT_SUPPORTED
cn-northwest-1:
  HVM64: ami-6b6a7d09
  HVMG2: NOT_SUPPORTED
Resources:
  CloudWatchPutMetricsRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
        Path: "/"
  CloudWatchPutMetricsRolePolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: CloudWatch_PutMetricData
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: CloudWatchPutMetricData
            Effect: Allow
            Action:
              - cloudwatch:PutMetricData
            Resource:
              - "*"
      Roles:
        - Ref: CloudWatchPutMetricsRole
  CloudWatchPutMetricsInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - Ref: CloudWatchPutMetricsRole
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      EnableDnsSupport: 'true'
```

```
    EnableDnsHostnames: 'true'
    CidrBlock: 10.0.0.0/16
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Application
          Value:
            Ref: AWS::StackName
        - Key: Network
          Value: Public
  GatewayToInternet:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId:
        Ref: VPC
      InternetGatewayId:
        Ref: InternetGateway
  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId:
        Ref: VPC
  SubnetRouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId:
        Ref: RouteTable
      SubnetId:
        Ref: Subnet
  InternetGatewayRoute:
    Type: AWS::EC2::Route
    Properties:
      DestinationCidrBlock: 0.0.0.0/0
      RouteTableId:
        Ref: RouteTable
      GatewayId:
        Ref: InternetGateway
  Subnet:
    Type: AWS::EC2::Subnet
```

Properties:**VpcId:**

Ref: VPC

CidrBlock: 10.0.0.0/24

Tags:

- Key: Application

Value:

Ref: AWS::StackId

InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:**VpcId:**

Ref: VPC

GroupDescription: Enable SSH access via port 22

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp:

Ref: SSHLocation

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

MountTargetSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:**VpcId:**

Ref: VPC

GroupDescription: Security group for mount target

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 2049

ToPort: 2049

CidrIp: 0.0.0.0/0

FileSystem:

Type: AWS::EFS::FileSystem

Properties:

PerformanceMode: generalPurpose

FileSystemTags:

- Key: Name

Value:

Ref: VolumeName

MountTarget:

```

Type: AWS::EFS::MountTarget
Properties:
  FileSystemId:
    Ref: FileSystem
  SubnetId:
    Ref: Subnet
  SecurityGroups:
    - Ref: MountTargetSecurityGroup
LaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        MountConfig:
          - setup
          - mount
      setup:
        packages:
          yum:
            nfs-utils: []
        files:
          "/home/ec2-user/post_nfsstat":
            content: !Sub |
              #!/bin/bash

              INPUT="$(cat)"
              CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
              CW_JSON_CLOSE=' ] }'
              CW_JSON_METRIC=''
              METRIC_COUNTER=0

              for COL in 1 2 3 4 5 6; do

                COUNTER=0
                METRIC_FIELD=$COL
                DATA_FIELD=$(( $COL + ( $COL - 1 ) ))

                while read line; do
                  if [[ COUNTER -gt 0 ]]; then

                    LINE=`echo $line | tr -s ' ' `
                    AWS_COMMAND="aws cloudwatch put-metric-data --region
${AWS::Region}"

                    MOD=$(( $COUNTER % 2))

```

```

        if [ $MOD -eq 1 ]; then
            METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
        else
            METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
        fi

        if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
            INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)
            CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"

            unset METRIC_NAME
            unset METRIC_VALUE

            METRIC_COUNTER=$((METRIC_COUNTER+1))
            if [ $METRIC_COUNTER -eq 20 ]; then
                # 20 is max metric collection size, so we have to submit here
                aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                # reset
                METRIC_COUNTER=0
                CW_JSON_METRIC=''
            fi
        fi

        COUNTER=$((COUNTER+1))
    fi

    if [[ "$line" == "Client nfs v4:" ]]; then
        # the next line is the good stuff
        COUNTER=$((COUNTER+1))
    fi
done <<< "$INPUT"
done

# submit whatever is left
aws cloudwatch put-metric-data --region ${AWS::Region} --cli-input-json
"`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"
mode: '000755'

```

```

        owner: ec2-user
        group: ec2-user
    "/home/ec2-user/crontab":
        content: "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
        owner: ec2-user
        group: ec2-user
    commands:
        01_createdir:
            command: !Sub "mkdir /${MountPoint}"
    mount:
        commands:
            01_mount:
                command: !Sub >
                    mount -t nfs4 -o nfsvers=4.1 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}
            02_permissions:
                command: !Sub "chown ec2-user:ec2-user /${MountPoint}"
Properties:
    AssociatePublicIpAddress: true
    ImageId:
        Fn::FindInMap:
            - AWSRegionArch2AMI
            - Ref: AWS::Region
        Fn::FindInMap:
            - AWSInstanceType2Arch
            - Ref: InstanceType
            - Arch
    InstanceType:
        Ref: InstanceType
    KeyName:
        Ref: KeyName
    SecurityGroups:
        - Ref: InstanceSecurityGroup
    IamInstanceProfile:
        Ref: CloudWatchPutMetricsInstanceProfile
    UserData:
        Fn::Base64: !Sub |
            #!/bin/bash -xe
            yum install -y aws-cfn-bootstrap
            /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchConfiguration --configsets MountConfig --region ${AWS::Region}
            crontab /home/ec2-user/crontab
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
AutoScalingGroup --region ${AWS::Region}

```



```
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - MountTarget
    - GatewayToInternet
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
    Count:
      Ref: AsgMaxSize
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    LaunchConfigurationName:
      Ref: LaunchConfiguration
    MinSize: '1'
    MaxSize:
      Ref: AsgMaxSize
    DesiredCapacity:
      Ref: AsgMaxSize
    Tags:
      - Key: Name
        Value: EFS FileSystem Mounted Instance
        PropagateAtLaunch: 'true'
  Outputs:
    MountTargetID:
      Description: Mount target ID
      Value:
        Ref: MountTarget
    FileSystemID:
      Description: File system ID
      Value:
        Ref: FileSystem
```

Extraits de modèle Elastic Beanstalk

Avec Elastic Beanstalk, vous pouvez rapidement déployer et gérer des AWS applications sans vous soucier de l'infrastructure qui exécute ces applications. L'exemple de modèle suivant peut vous aider à décrire les ressources Elastic Beanstalk de votre modèle. CloudFormation

Exemple d'application PHP pour Elastic Beanstalk

L'exemple de modèle suivant déploie un exemple d'application web PHP stocké dans un compartiment Amazon S3. L'environnement est également un environnement d'auto-scaling et d'équilibrage de charge, avec un minimum de deux EC2 instances Amazon et un maximum de six. Il montre un environnement Elastic Beanstalk qui utilise une ancienne configuration de lancement. Pour plus d'informations sur l'utilisation d'un modèle de lancement, consultez la section [Modèles de lancement](#) du Guide du développeur AWS Elastic Beanstalk .

Remplacez *solution-stack* par le nom d'une pile de solution (version de plateforme). Pour obtenir la liste des piles de solutions disponibles, utilisez la AWS CLI commande `aws elasticbeanstalk list-available-solution-stacks`.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "sampleApplication": {
      "Type": "AWS::ElasticBeanstalk::Application",
      "Properties": {
        "Description": "AWS Elastic Beanstalk Sample Application"
      }
    },
    "sampleApplicationVersion": {
      "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
      "Properties": {
        "ApplicationName": {
          "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Application Version",
        "SourceBundle": {
          "S3Bucket": {
            "Fn::Sub": "elasticbeanstalk-samples-${AWS::Region}"
          },
          "S3Key": "php-newsample-app.zip"
        }
      }
    },
    "sampleConfigurationTemplate": {
      "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
      "Properties": {
```

```

    "ApplicationName": {
      "Ref": "sampleApplication"
    },
    "Description": "AWS ElasticBeanstalk Sample Configuration Template",
    "OptionSettings": [
      {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MinSize",
        "Value": "2"
      },
      {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MaxSize",
        "Value": "6"
      },
      {
        "Namespace": "aws:elasticbeanstalk:environment",
        "OptionName": "EnvironmentType",
        "Value": "LoadBalanced"
      },
      {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": {
          "Ref": "MyInstanceProfile"
        }
      }
    ],
    "SolutionStackName": "solution-stack"
  }
},
"sampleEnvironment": {
  "Type": "AWS::ElasticBeanstalk::Environment",
  "Properties": {
    "ApplicationName": {
      "Ref": "sampleApplication"
    },
    "Description": "AWS ElasticBeanstalk Sample Environment",
    "TemplateName": {
      "Ref": "sampleConfigurationTemplate"
    },
    "VersionLabel": {
      "Ref": "sampleApplicationVersion"
    }
  }
}

```

```

    }
  },
  "MyInstanceRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "ec2.amazonaws.com"
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ]
          }
        ]
      },
      "Description": "Beanstalk EC2 role",
      "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier",
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker",
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier"
      ]
    }
  },
  "MyInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
      "Roles": [
        {
          "Ref": "MyInstanceRole"
        }
      ]
    }
  }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  sampleApplication:
    Type: AWS::ElasticBeanstalk::Application
    Properties:
      Description: AWS Elastic Beanstalk Sample Application
  sampleApplicationVersion:
    Type: AWS::ElasticBeanstalk::ApplicationVersion
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Application Version
      SourceBundle:
        S3Bucket: !Sub "elasticbeanstalk-samples-${AWS::Region}"
        S3Key: php-newsample-app.zip
  sampleConfigurationTemplate:
    Type: AWS::ElasticBeanstalk::ConfigurationTemplate
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Configuration Template
      OptionSettings:
        - Namespace: aws:autoscaling:asg
          OptionName: MinSize
          Value: '2'
        - Namespace: aws:autoscaling:asg
          OptionName: MaxSize
          Value: '6'
        - Namespace: aws:elasticbeanstalk:environment
          OptionName: EnvironmentType
          Value: LoadBalanced
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile
          Value: !Ref MyInstanceProfile
      SolutionStackName: solution-stack
  sampleEnvironment:
    Type: AWS::ElasticBeanstalk::Environment
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Environment
      TemplateName:
```

```
    Ref: sampleConfigurationTemplate
  VersionLabel:
    Ref: sampleApplicationVersion
MyInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Description: Beanstalk EC2 role
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier
MyInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref MyInstanceRole
```

Extraits de modèle Elastic Load Balancing

Pour créer un Application Load Balancer, un Network Load Balancer ou un Gateway Load Balancer, utilisez les types de ressources V2, qui commencent par `AWS::ElasticLoadBalancingV2`.

Pour créer un Classic Load Balancer, utilisez les types de ressources commençant par `AWS::ElasticLoadBalancing`.

Table des matières

- [ELBv2 ressources](#)
- [Ressources Classic Load Balancer](#)

ELBv2 ressources

Cet exemple définit un Application Load Balancer doté d'un écouteur HTTP et d'une action par défaut qui transmet le trafic au groupe cible. L'équilibreur de charge classique utilise les paramètres de surveillance de l'état par défaut. Le groupe cible possède deux EC2 instances enregistrées.

YAML

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myHTTPListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup

  myTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      Name: "my-target-group"
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
```

Port: 80

JSON

```
{
  "Resources": {
    "myLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
        "Name": "my-alb",
        "Type": "application",
        "Scheme": "internal",
        "Subnets": [
          {
            "Ref": "subnet-AZ1"
          },
          {
            "Ref": "subnet-AZ2"
          }
        ],
        "SecurityGroups": [
          {
            "Ref": "mySecurityGroup"
          }
        ]
      }
    },
    "myHTTPListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "LoadBalancerArn": {
          "Ref": "myLoadBalancer"
        },
        "Protocol": "HTTP",
        "Port": 80,
        "DefaultActions": [
          {
            "Type": "forward",
            "TargetGroupArn": {
              "Ref": "myTargetGroup"
            }
          }
        ]
      }
    }
  }
}
```



```

    }
  },
  "myTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "Name": "my-target-group",
      "Protocol": "HTTP",
      "Port": 80,
      "TargetType": "instance",
      "VpcId": {
        "Ref": "myVPC"
      },
      "Targets": [
        {
          "Id": {
            "Fn::GetAtt": [
              "Instance1",
              "InstanceId"
            ]
          },
          "Port": 80
        },
        {
          "Id": {
            "Fn::GetAtt": [
              "Instance2",
              "InstanceId"
            ]
          },
          "Port": 80
        }
      ]
    }
  }
}

```

Ressources Classic Load Balancer

Cet exemple définit un Classic Load Balancer avec un écouteur HTTP et aucune instance enregistrée. EC2 L'équilibreur de charge classique utilise les paramètres de surveillance de l'état par défaut.

YAML

```
myLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
```

JSON

```
"myLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
}
```

Cet exemple définit un Classic Load Balancer avec un écouteur HTTP, deux EC2 instances enregistrées et des paramètres de contrôle de santé personnalisés.

YAML

```
myClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Instances:
      - Ref: Instance1
      - Ref: Instance2
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
```

```
Protocol: HTTP
HealthCheck:
  Target: HTTP:80/
  HealthyThreshold: '3'
  UnhealthyThreshold: '5'
  Interval: '30'
  Timeout: '5'
```

JSON

```
"myClassicLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Instances" : [
      { "Ref" : "Instance1" },
      { "Ref" : "Instance2" }
    ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
}
```

Gestion des identités et des accès AWS extraits de modèles

Cette section contient des extraits Gestion des identités et des accès AWS de modèles.

Rubriques

- [Déclaration d'une ressource d'utilisateur IAM](#)
- [Déclaration d'une ressource de clé d'accès IAM](#)

- [Déclaration d'une ressource de groupe IAM](#)
- [Ajout d'utilisateurs à un groupe](#)
- [Déclaration d'une politique IAM](#)
- [Déclaration d'une politique de compartiment Amazon S3](#)
- [Déclaration d'une politique de rubrique Amazon SNS](#)
- [Déclaration d'une politique Amazon SQS](#)
- [Exemples de modèle pour les rôles IAM](#)

Important

Lorsque vous créez ou mettez à jour une pile à l'aide d'un modèle contenant des ressources IAM, vous devez confirmer l'utilisation des fonctionnalités IAM. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).

Déclaration d'une ressource d'utilisateur IAM

Cet extrait de code présente comment déclarer une ressource [AWS::IAM::User](#) pour créer un utilisateur IAM. L'utilisateur est déclaré avec le chemin ("/") et un profil de connexion utilisant le mot de passe (myP@ssW0rd).

Le document de politique intitulé `giveaccesstoqueueonly` accorde à l'utilisateur l'autorisation d'effectuer toutes les actions Amazon SQS au niveau de la ressource de file d'attente Amazon SQS `myqueue`, et refuse l'accès à toutes les autres ressources de file d'attente Amazon SQS. La fonction `Fn::GetAtt` obtient l'attribut `Arn` de la ressource [AWS::SQS::Queue](#) `myqueue`.

Le document de politique intitulé `giveaccesstotopiconly` est ajouté à l'utilisateur pour l'autoriser à effectuer toutes les actions Amazon SNS au niveau de la ressource de rubrique Amazon SNS `mytopic`, et refuser l'accès à toutes les autres ressources Amazon SNS. La fonction `Ref` obtient l'ARN de la [AWS::SNS::Topic](#) ressource `mytopic`.

JSON

```
"myuser" : {
  "Type" : "AWS::IAM::User",
  "Properties" : {
```

```

    "Path" : "/",
    "LoginProfile" : {
        "Password" : "myP@ssW0rd"
    },
    "Policies" : [ {
        "PolicyName" : "giveaccesstoqueueonly",
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
                "Effect" : "Allow",
                "Action" : [ "sqs:*" ],
                "Resource" : [ {
                    "Fn::GetAtt" : [ "myqueue", "Arn" ]
                } ]
            }, {
                "Effect" : "Deny",
                "Action" : [ "sqs:*" ],
                "NotResource" : [ {
                    "Fn::GetAtt" : [ "myqueue", "Arn" ]
                } ]
            }
        ]
    }, {
        "PolicyName" : "giveaccesstotopiconly",
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
                "Effect" : "Allow",
                "Action" : [ "sns:*" ],
                "Resource" : [ { "Ref" : "mytopic" } ]
            }, {
                "Effect" : "Deny",
                "Action" : [ "sns:*" ],
                "NotResource" : [ { "Ref" : "mytopic" } ]
            } ]
        ]
    } ]
}

```

YAML

```
myuser:
```

```
Type: AWS::IAM::User
Properties:
  Path: "/"
  LoginProfile:
    Password: myP@ssW0rd
  Policies:
    - PolicyName: giveaccesstoqueueonly
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - sqs:*
            Resource:
              - !GetAtt myqueue.Arn
          - Effect: Deny
            Action:
              - sqs:*
            NotResource:
              - !GetAtt myqueue.Arn
    - PolicyName: giveaccesstotopiconly
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - sns:*
            Resource:
              - !Ref mytopic
          - Effect: Deny
            Action:
              - sns:*
            NotResource:
              - !Ref mytopic
```

Déclaration d'une ressource de clé d'accès IAM

Cet extrait illustre une ressource [AWS::IAM::AccessKey](#). La ressource myaccesskey crée une clé d'accès et l'attribue à un utilisateur IAM qui est déclaré en tant que ressource [AWS::IAM::User](#) dans le modèle.

JSON

```
"myaccesskey" : {
  "Type" : "AWS::IAM::AccessKey",
  "Properties" : {
    "UserName" : { "Ref" : "myuser" }
  }
}
```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

Vous pouvez obtenir la clé secrète d'une ressource `AWS::IAM::AccessKey` à l'aide de la fonction `Fn::GetAtt`. L'un des moyens de récupérer la clé secrète est de la placer dans une valeur `Output`. Vous pouvez obtenir la clé d'accès à l'aide de la fonction `Ref`. Les déclarations de valeurs `Output` suivantes obtiennent la clé d'accès et la clé secrète pour `myaccesskey`.

JSON

```
"AccessKeyformyaccesskey" : {
  "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
  "Value" : {
    "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
  }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

Vous pouvez également transmettre la clé AWS d'accès et la clé secrète à une EC2 instance Amazon ou à un groupe Auto Scaling défini dans le modèle. La déclaration [AWS::EC2::Instance](#) suivante utilise la propriété UserData pour transmettre la clé d'accès et la clé secrète pour la ressource myaccesskey.

JSON

```
"myinstance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "AvailabilityZone" : "us-east-1a",
    "ImageId" : "ami-0ff8a91507f77f867",
    "UserData" : {
      "Fn::Base64" : {
        "Fn::Join" : [
          "", [
            "ACCESS_KEY=", {
              "Ref" : "myaccesskey"
            },
            "&",
            "SECRET_KEY=",
            {
              "Fn::GetAtt" : [
                "myaccesskey",
                "SecretAccessKey"
              ]
            }
          ]
        ]
      }
    }
  }
}
```

YAML

```
myinstance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: "us-east-1a"
    ImageId: ami-0ff8a91507f77f867
    UserData:
```



```
Fn::Base64: !Sub "ACCESS_KEY=${myaccesskey}&SECRET_KEY=
${myaccesskey.SecretAccessKey}"
```

Déclaration d'une ressource de groupe IAM

Cet extrait illustre une ressource [AWS::IAM::Group](#). Le groupe a un chemin ("/myapplication/"). Le document de politique intitulé myapppolicy est ajouté au groupe pour autoriser les utilisateurs du groupe à effectuer toutes les actions Amazon SQS au niveau de la ressource de file d'attente Amazon SQS myqueue, et refuser l'accès à toutes les autres ressources Amazon SQS à l'exception de myqueue.

Pour attribuer une politique à une ressource, IAM nécessite l'Amazon Resource Name (ARN) de la ressource. Dans l'extrait, la fonction `Fn::GetAtt` obtient l'ARN de la file d'attente de la ressource [AWS::SQS::Queue](#).

JSON

```
"mygroup" : {
  "Type" : "AWS::IAM::Group",
  "Properties" : {
    "Path" : "/myapplication/",
    "Policies" : [ {
      "PolicyName" : "myapppolicy",
      "PolicyDocument" : {
        "Version": "2012-10-17",
        "Statement" : [ {
          "Effect" : "Allow",
          "Action" : [ "sqs:*" ],
          "Resource" : [ {
            "Fn::GetAtt" : [ "myqueue", "Arn" ]
          } ]
        } ],
      },
      {
        "Effect" : "Deny",
        "Action" : [ "sqs:*" ],
        "NotResource" : [ { "Fn::GetAtt" : [ "myqueue", "Arn" ] } ]
      }
    ] ]
  } ]
}
```

YAML

```
mygroup:
  Type: AWS::IAM::Group
  Properties:
    Path: "/myapplication/"
    Policies:
      - PolicyName: myapppolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - sqs:*
              Resource: !GetAtt myqueue.Arn
            - Effect: Deny
              Action:
                - sqs:*
              NotResource: !GetAtt myqueue.Arn
```

Ajout d'utilisateurs à un groupe

La ressource [AWS::IAM::UserToGroupAddition](#) ajoute des utilisateurs à un groupe. Dans l'extrait suivant, la ressource `addUserToGroup` ajoute les utilisateurs suivants à un groupe existant nommé `myexistinggroup2` : l'utilisateur existant `existinguser1` et l'utilisateur `myuser` qui est déclaré comme une ressource [AWS::IAM::User](#) dans le modèle.

JSON

```
"addUserToGroup" : {
  "Type" : "AWS::IAM::UserToGroupAddition",
  "Properties" : {
    "GroupName" : "myexistinggroup2",
    "Users" : [ "existinguser1", { "Ref" : "myuser" } ]
  }
}
```

YAML

```
addUserToGroup:
  Type: AWS::IAM::UserToGroupAddition
  Properties:
```

```

GroupName: myexistinggroup2
Users:
- existinguser1
- !Ref myuser

```

Déclaration d'une politique IAM

Cet extrait de code montre comment créer une politique et l'appliquer à plusieurs groupes à l'aide d'une ressource [AWS::IAM::Policy](#) nommée `mypolicy`. La ressource `mypolicy` contient une propriété `PolicyDocument` qui autorise les actions `GetObject`, `PutObject` et `PutObjectAcl` sur les objets du compartiment S3 représenté par l'ARN `arn:aws:s3:::myAWSBucket`. La ressource `mypolicy` applique la politique à un groupe existant nommé `myexistinggroup1` et à un groupe `mygroup` déclaré dans le modèle en tant que ressource [AWS::IAM::Group](#). Cet exemple montre comment appliquer une politique à un groupe à l'aide de la propriété `Groups`. Toutefois, vous pouvez aussi utiliser la propriété `Users` pour ajouter un document de politique à une liste d'utilisateurs.

JSON

```

"mypolicy" : {
  "Type" : "AWS::IAM::Policy",
  "Properties" : {
    "PolicyName" : "mygrouppolicy",
    "PolicyDocument" : {
      "Version": "2012-10-17",
      "Statement" : [ {
        "Effect" : "Allow",
        "Action" : [
          "s3:GetObject" , "s3:PutObject" , "s3:PutObjectAcl" ],
        "Resource" : "arn:aws:s3:::myAWSBucket/*"
      } ]
    },
    "Groups" : [ "myexistinggroup1", { "Ref" : "mygroup" } ]
  }
}

```

YAML

```

mypolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: mygrouppolicy
    PolicyDocument:

```

```

Version: '2012-10-17'
Statement:
- Effect: Allow
  Action:
  - s3:GetObject
  - s3:PutObject
  - s3:PutObjectAcl
  Resource: arn:aws:s3:::myAWSBucket/*
Groups:
- myexistinggroup1
- !Ref mygroup

```

Déclaration d'une politique de compartiment Amazon S3

Cet extrait de code montre comment créer une politique et l'appliquer à un compartiment Amazon S3 à l'aide de la ressource [AWS::S3::BucketPolicy](#). La ressource `mybucketpolicy` déclare un document de politique qui autorise l'utilisateur IAM `user1` à effectuer l'action `GetObject` sur tous les objets du compartiment S3 auquel cette politique est appliquée. Dans l'extrait, la fonction `Fn::GetAtt` obtient l'ARN de la ressource `user1`. La ressource `mybucketpolicy` applique la politique à la ressource `AWS::S3::BucketPolicy` intitulée `mybucket`. La fonction `Ref` récupère le nom de compartiment de la ressource `mybucket`.

JSON

```

"mybucketpolicy" : {
  "Type" : "AWS::S3::BucketPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "ReadAccess",
        "Action" : [ "s3:GetObject" ],
        "Effect" : "Allow",
        "Resource" : { "Fn::Join" : [
          "", [ "arn:aws:s3:::", { "Ref" : "mybucket" } , "/*" ]
        ] },
        "Principal" : {
          "AWS" : { "Fn::GetAtt" : [ "user1", "Arn" ] }
        }
      ]
    }
  },

```

```
    "Bucket" : { "Ref" : "mybucket" }
  }
}
```

YAML

```
mybucketpolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    PolicyDocument:
      Id: MyPolicy
      Version: '2012-10-17'
      Statement:
        - Sid: ReadAccess
          Action:
            - s3:GetObject
          Effect: Allow
          Resource: !Sub "arn:aws:s3:::${mybucket}/*"
          Principal:
            AWS: !GetAtt user1.Arn
      Bucket: !Ref mybucket
```

Déclaration d'une politique de rubrique Amazon SNS

Cet extrait de code montre comment créer une politique et l'appliquer à une rubrique Amazon SNS à l'aide de la ressource [AWS::SNS::TopicPolicy](#). La ressource `mysnspolicy` contient une propriété `PolicyDocument` qui permet à la ressource [AWS::IAM::User](#) intitulée `myuser` d'effectuer l'action `Publish` sur une ressource [AWS::SNS::Topic](#) intitulée `mytopic`. Dans cet extrait, la fonction `Fn::GetAtt` obtient l'ARN de la ressource `myuser`, tandis que la fonction `Ref` obtient l'ARN de la ressource `mytopic`.

JSON

```
"mysnspolicy" : {
  "Type" : "AWS::SNS::TopicPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyTopicPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "My-statement-id",
        "Effect" : "Allow",
```

```

        "Principal" : {
            "AWS" : { "Fn::GetAtt" : [ "myuser", "Arn" ] }
        },
        "Action" : "sns:Publish",
        "Resource" : "*"
    } ]
},
"Topics" : [ { "Ref" : "mytopic" } ]
}
}

```

YAML

```

mysnspolicy:
  Type: AWS::SNS::TopicPolicy
  Properties:
    PolicyDocument:
      Id: MyTopicPolicy
      Version: '2012-10-17'
      Statement:
        - Sid: My-statement-id
          Effect: Allow
          Principal:
            AWS: !GetAtt myuser.Arn
          Action: sns:Publish
          Resource: "*"
    Topics:
      - !Ref mytopic

```

Déclaration d'une politique Amazon SQS

Cet extrait de code montre comment créer une politique et l'appliquer à une file d'attente Amazon SQS à l'aide de la ressource [AWS::SQS::QueuePolicy](#). La propriété `PolicyDocument` permet à l'utilisateur existant `myapp` (spécifié par son ARN) d'effectuer l'action `SendMessage` sur une file d'attente existante, qui est spécifiée par son URL, et sur une ressource [AWS::SQS::Queue](#) intitulée `myqueue`. La fonction [Ref](#) obtient l'URL de la ressource `myqueue`.

JSON

```

"mysqspolicy" : {
  "Type" : "AWS::SQS::QueuePolicy",
  "Properties" : {

```

```

    "PolicyDocument" : {
      "Id" : "MyQueuePolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "Allow-User-SendMessage",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : "arn:aws:iam::123456789012:user/myapp"
        },
        "Action" : [ "sqs:SendMessage" ],
        "Resource" : "*"
      } ]
    },
    "Queues" : [
      "https://sqs.us-east-2aws-region.amazonaws.com/123456789012/myexistingqueue",
      { "Ref" : "myqueue" }
    ]
  }
}

```

YAML

```

mysqspolicy:
  Type: AWS::SQS::QueuePolicy
  Properties:
    PolicyDocument:
      Id: MyQueuePolicy
      Version: '2012-10-17'
      Statement:
        - Sid: Allow-User-SendMessage
          Effect: Allow
          Principal:
            AWS: arn:aws:iam::123456789012:user/myapp
          Action:
            - sqs:SendMessage
          Resource: "*"
    Queues:
      - https://sqs.aws-region.amazonaws.com/123456789012/myexistingqueue
      - !Ref myqueue

```

Exemples de modèle pour les rôles IAM

Cette section fournit des exemples de CloudFormation modèles de rôles IAM pour les EC2 instances.

Pour plus d'informations, consultez la section [Rôles IAM pour Amazon EC2](#) dans le guide de l' EC2 utilisateur Amazon.

Rôle IAM auprès de EC2

Dans cet exemple, le profil d'instance est référencé par la `IamInstanceProfile` propriété de l' EC2 instance. La politique de l'instance et la politique du rôle référencent toutes les deux la ressource [AWS::IAM::Role](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "myEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Version": "2009-05-15",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "m1.small",
        "Monitoring": "true",
        "DisableApiTermination": "false",
        "IamInstanceProfile": {
          "Ref": "RootInstanceProfile"
        }
      }
    },
    "RootRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [ {
            "Effect": "Allow",
            "Principal": {
              "Service": [ "ec2.amazonaws.com" ]
            },
            "Action": [ "sts:AssumeRole" ]
          } ]
        }
      }
    },
    "Path": "/"
  },
}
```



```

    "RolePolicies": {
      "Type": "AWS::IAM::Policy",
      "Properties": {
        "PolicyName": "root",
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [ {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
          } ]
        },
        "Roles": [ { "Ref": "RootRole" } ]
      }
    },
    "RootInstanceProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Properties": {
        "Path": "/",
        "Roles": [ { "Ref": "RootRole" } ]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      Monitoring: 'true'
      DisableApiTermination: 'false'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:

```

```
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service:
            - ec2.amazonaws.com
        Action:
          - sts:AssumeRole
    Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action: "*"
            Resource: "*"
      Roles:
        - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - !Ref RootRole
```

Rôle IAM avec groupe Auto Scaling

Dans cet exemple, le profil d'instance est référencé par la `IamInstanceProfile` propriété d'une configuration de lancement Amazon EC2 Auto Scaling.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "myLCOne": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Version": "2009-05-15",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
```

```

        "InstanceType": "m1.small",
        "InstanceMonitoring": "true",
        "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
    },
    "myASGrpOne": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "Version": "2009-05-15",
        "Properties": {
            "AvailabilityZones": [ "us-east-1a" ],
            "LaunchConfigurationName": { "Ref": "myLCOne" },
            "MinSize": "0",
            "MaxSize": "0",
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": "120"
        }
    },
    "RootRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [ {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [ "ec2.amazonaws.com" ]
                    },
                    "Action": [ "sts:AssumeRole" ]
                } ]
            },
            "Path": "/"
        }
    },
    "RolePolicies": {
        "Type": "AWS::IAM::Policy",
        "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [ {
                    "Effect": "Allow",
                    "Action": "*",
                    "Resource": "*"
                } ]
            }
        }
    }
}

```

```

        },
        "Roles": [ { "Ref": "RootRole" } ]
    }
},
"RootInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [ { "Ref": "RootRole" } ]
    }
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myLCOne:
    Type: AWS::AutoScaling::LaunchConfiguration
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      InstanceMonitoring: 'true'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  myASGrpOne:
    Type: AWS::AutoScaling::AutoScalingGroup
    Version: '2009-05-15'
    Properties:
      AvailabilityZones:
        - "us-east-1a"
      LaunchConfigurationName:
        !Ref myLCOne
      MinSize: '0'
      MaxSize: '0'
      HealthCheckType: EC2
      HealthCheckGracePeriod: '120'
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:

```

```
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service:
            - ec2.amazonaws.com
        Action:
          - sts:AssumeRole
    Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action: "*"
            Resource: "*"
      Roles:
        - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - !Ref RootRole
```

AWS Lambda modèle

Le modèle suivant utilise une fonction AWS Lambda (Lambda) et une ressource personnalisée pour ajouter un nouveau groupe de sécurité à une liste de groupes de sécurité existants. Cette fonction s'avère utile lorsque vous voulez créer dynamiquement une liste de groupes de sécurité, afin d'y inclure des groupes de sécurité à la fois nouveaux et existants. Par exemple, vous pouvez transmettre une liste de groupes de sécurité existants sous forme de valeur de paramètre, ajouter la nouvelle valeur à la liste, puis associer toutes vos valeurs à une EC2 instance. Pour plus d'informations sur le type de ressource de fonction Lambda, consultez [AWS::Lambda::Function](#).

Dans l'exemple, lors de la CloudFormation création de la ressource AllSecurityGroups personnalisée, CloudFormation invoque la fonction AppendItemToListFunction Lambda. CloudFormation transmet la liste des groupes de sécurité existants et un nouveau groupe de sécurité (NewSecurityGroup) à la fonction, qui ajoute le nouveau groupe de sécurité à la liste puis renvoie

la liste modifiée. CloudFormation utilise la liste modifiée pour associer tous les groupes de sécurité à la MyEC2Instance ressource.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ExistingSecurityGroups": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>"
    },
    "ExistingVPC": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "The VPC ID that includes the security groups in the ExistingSecurityGroups parameter."
    },
    "InstanceType": {
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": [
        "t2.micro",
        "t3.micro"
      ]
    }
  },
  "Resources": {
    "SecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Allow HTTP traffic to the host",
        "VpcId": {
          "Ref": "ExistingVPC"
        }
      },
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        }
      ],
      "SecurityGroupEgress": [
```

```

        {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIp": "0.0.0.0/0"
        }
    ]
},
"AllSecurityGroups": {
    "Type": "Custom::Split",
    "Properties": {
        "ServiceToken": {
            "Fn::GetAtt": [
                "AppendItemToListFunction",
                "Arn"
            ]
        },
        "List": {
            "Ref": "ExistingSecurityGroups"
        },
        "AppendedItem": {
            "Ref": "SecurityGroup"
        }
    }
},
"AppendItemToListFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
        "Handler": "index.handler",
        "Role": {
            "Fn::GetAtt": [
                "LambdaExecutionRole",
                "Arn"
            ]
        },
        "Code": {
            "ZipFile": {
                "Fn::Join": [
                    "",
                    [
                        "var response = require('cfn-response');",
                        "exports.handler = function(event, context) {"

```

```

        "    var responseData = {Value:
event.ResourceProperties.List}};",
        "
responseData.Value.push(event.ResourceProperties.AppendedItem);",
        "    response.send(event, context, response.SUCCESS,
responseData);",
        "};"
    ]
}
},
"Runtime": "nodejs20.x"
}
},
"MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-
ami-hvm-x86_64-gp2}}",
        "SecurityGroupIds": {
            "Fn::GetAtt": [
                "AllSecurityGroups",
                "Value"
            ]
        },
        "InstanceType": {
            "Ref": "InstanceType"
        }
    }
},
"LambdaExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lambda.amazonaws.com"
                        ]
                    },
                    "Action": [

```



```

        "sts:AssumeRole"
      ]
    }
  ]
},
"Path": "/",
"Policies": [
  {
    "PolicyName": "root",
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "logs:*"
          ],
          "Resource": "arn:aws:logs:*:*:*"
        }
      ]
    }
  }
]
}
}
},
"Outputs": {
  "AllSecurityGroups": {
    "Description": "Security Groups that are associated with the EC2 instance",
    "Value": {
      "Fn::Join": [
        ",",
        [
          {
            "Fn::GetAtt": [
              "AllSecurityGroups",
              "Value"
            ]
          }
        ]
      ]
    }
  }
}
}
}
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  ExistingSecurityGroups:
    Type: List<AWS::EC2::SecurityGroup::Id>
  ExistingVPC:
    Type: AWS::EC2::VPC::Id
    Description: The VPC ID that includes the security groups in the
ExistingSecurityGroups parameter.
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP traffic to the host
      VpcId: !Ref ExistingVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
  AllSecurityGroups:
    Type: Custom::Split
    Properties:
      ServiceToken: !GetAtt AppendItemToListFunction.Arn
      List: !Ref ExistingSecurityGroups
      AppendedItem: !Ref SecurityGroup
  AppendItemToListFunction:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
```

```

Code:
  ZipFile: !Join
    - ''
    - - var response = require('cfn-response');
      - exports.handler = function(event, context) {
      - '   var responseData = {Value: event.ResourceProperties.List};'
      - '   responseData.Value.push(event.ResourceProperties.AppendedItem);'
      - '   response.send(event, context, response.SUCCESS, responseData);'
      - '};'
  Runtime: nodejs20.x
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    SecurityGroupIds: !GetAtt AllSecurityGroups.Value
    InstanceType: !Ref InstanceType
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: /
    Policies:
      - PolicyName: root
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - logs:*
              Resource: arn:aws:logs:*:*:*
Outputs:
  AllSecurityGroups:
    Description: Security Groups that are associated with the EC2 instance
    Value: !Join
      - ', '

```

```
- !GetAtt AllSecurityGroups.Value
```

Extraits de modèle Amazon Redshift

Amazon Redshift est un service d'entrepôt des données entièrement géré dans le cloud. Vous pouvez utiliser CloudFormation pour allouer et gérer des clusters Amazon Redshift.

Cluster Amazon Redshift

L'exemple de modèle suivant crée un cluster Amazon Redshift selon les valeurs des paramètres spécifiées lors de la création de la pile. Le groupe de paramètres du cluster associé au cluster Amazon Redshift permet de journaliser l'activité des utilisateurs. Le modèle lance également les clusters Amazon Redshift dans un Amazon VPC qui est défini dans le modèle. Le VPC comporte une passerelle internet afin que vous puissiez accéder aux clusters Amazon Redshift à partir d'Internet. Toutefois, la communication entre le cluster et la passerelle Internet doit aussi être activée, ce qui est effectuée par l'entrée de la table de routage.

Note

Le modèle inclut la condition `IsMultiNodeCluster` pour que le paramètre `NumberOfNodes` soit déclaré uniquement lorsque la valeur de paramètre `ClusterType` indique `multi-node`.

L'exemple définit le paramètre `MysqlRootPassword` avec sa propriété `NoEcho` définie sur `true`. Si vous définissez l'attribut `NoEcho` sur `true`, CloudFormation renvoie la valeur du paramètre masquée sous forme d'astérisques (`*****`) pour tous les appels qui décrivent la pile ou les événements de pile, à l'exception des informations stockées dans les emplacements spécifiés ci-dessous.

Important

L'utilisation de l'attribut `NoEcho` ne masque aucune information stockée dans les lieux suivants :

- La section des `Metadata` modèles. CloudFormation ne transforme, ne modifie ni n'expurge aucune information que vous incluez dans `Metadata` cette section. Pour de plus amples informations, veuillez consulter [Métadonnées](#).
- La section de modèle `Outputs` Pour de plus amples informations, veuillez consulter [Sorties](#)

- L'attribut Metadata d'une définition de ressource. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Nous vous recommandons vivement de ne pas utiliser ces mécanismes pour inclure des informations sensibles, telles que des mots de passe ou des secrets.

Important

Plutôt que d'intégrer des informations sensibles directement dans vos CloudFormation modèles, nous vous recommandons d'utiliser des paramètres dynamiques dans le modèle de pile pour référencer les informations sensibles stockées et gérées à l'extérieur CloudFormation, par exemple dans le AWS Systems Manager Parameter Store ou AWS Secrets Manager.

Pour de plus amples informations, veuillez consulter la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the cluster is
created",
      "Type" : "String",
      "Default" : "dev",
      "AllowedPattern" : "([a-z]|[0-9])+"
    },
    "ClusterType" : {
      "Description" : "The type of cluster",
      "Type" : "String",
      "Default" : "single-node",
      "AllowedValues" : [ "single-node", "multi-node" ]
    },
    "NumberOfNodes" : {
      "Description" : "The number of compute nodes in the cluster. For multi-node
clusters, the NumberOfNodes parameter must be greater than 1",
      "Type" : "Number",
```

```

"Default" : "1"
},
"NodeType" : {
  "Description" : "The type of node to be provisioned",
  "Type" : "String",
  "Default" : "ds2.xlarge",
  "AllowedValues" : [ "ds2.xlarge", "ds2.8xlarge", "dc1.large", "dc1.8xlarge" ]
},
"MasterUsername" : {
  "Description" : "The user name that is associated with the master user account for the cluster that is being created",
  "Type" : "String",
  "Default" : "defaultuser",
  "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
},
"MasterUserPassword" : {
  "Description" : "The password that is associated with the master user account for the cluster that is being created.",
  "Type" : "String",
  "NoEcho" : "true"
},
"InboundTraffic" : {
  "Description" : "Allow inbound traffic to the cluster from this CIDR range.",
  "Type" : "String",
  "MinLength" : "9",
  "MaxLength" : "18",
  "Default" : "0.0.0.0/0",
  "AllowedPattern" : "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
  "ConstraintDescription" : "must be a valid CIDR range of the form x.x.x.x/x."
},
"PortNumber" : {
  "Description" : "The port number on which the cluster accepts incoming connections.",
  "Type" : "Number",
  "Default" : "5439"
}
},
"Conditions" : {
  "IsMultiNodeCluster" : {
    "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
  }
},
"Resources" : {

```

```

"RedshiftCluster" : {
  "Type" : "AWS::Redshift::Cluster",
  "DependsOn" : "AttachGateway",
  "Properties" : {
    "ClusterType" : { "Ref" : "ClusterType" },
    "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster", { "Ref" :
"NumberOfNodes" }, { "Ref" : "AWS::NoValue" } ]}],
    "NodeType" : { "Ref" : "NodeType" },
    "DBName" : { "Ref" : "DatabaseName" },
    "MasterUsername" : { "Ref" : "MasterUsername" },
    "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
    "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup" },
    "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
    "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
    "PubliclyAccessible" : "true",
    "Port" : { "Ref" : "PortNumber" }
  }
},
"RedshiftClusterParameterGroup" : {
  "Type" : "AWS::Redshift::ClusterParameterGroup",
  "Properties" : {
    "Description" : "Cluster parameter group",
    "ParameterGroupFamily" : "redshift-1.0",
    "Parameters" : [{
      "ParameterName" : "enable_user_activity_logging",
      "ParameterValue" : "true"
    }]
  }
},
"RedshiftClusterSubnetGroup" : {
  "Type" : "AWS::Redshift::ClusterSubnetGroup",
  "Properties" : {
    "Description" : "Cluster subnet group",
    "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
  }
},
"VPC" : {
  "Type" : "AWS::EC2::VPC",
  "Properties" : {
    "CidrBlock" : "10.0.0.0/16"
  }
},
"PublicSubnet" : {
  "Type" : "AWS::EC2::Subnet",

```

```

    "Properties" : {
      "CidrBlock" : "10.0.0.0/24",
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "SecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Security group",
      "SecurityGroupIngress" : [ {
        "CidrIp" : { "Ref" : "InboundTraffic" },
        "FromPort" : { "Ref" : "PortNumber" },
        "ToPort" : { "Ref" : "PortNumber" },
        "IpProtocol" : "tcp"
      } ],
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "myInternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway"
  },
  "AttachGateway" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "InternetGatewayId" : { "Ref" : "myInternetGateway" }
    }
  },
  "PublicRouteTable" : {
    "Type" : "AWS::EC2::RouteTable",
    "Properties" : {
      "VpcId" : {
        "Ref" : "VPC"
      }
    }
  },
  "PublicRoute" : {
    "Type" : "AWS::EC2::Route",
    "DependsOn" : "AttachGateway",
    "Properties" : {
      "RouteTableId" : {
        "Ref" : "PublicRouteTable"
      },
      "DestinationCidrBlock" : "0.0.0.0/0",

```



```

        "GatewayId" : {
            "Ref" : "myInternetGateway"
        }
    },
    "PublicSubnetRouteTableAssociation" : {
        "Type" : "AWS::EC2::SubnetRouteTableAssociation",
        "Properties" : {
            "SubnetId" : {
                "Ref" : "PublicSubnet"
            },
            "RouteTableId" : {
                "Ref" : "PublicRouteTable"
            }
        }
    },
    "Outputs" : {
        "ClusterEndpoint" : {
            "Description" : "Cluster endpoint",
            "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Port" ] } ] ] }
        },
        "ClusterName" : {
            "Description" : "Name of cluster",
            "Value" : { "Ref" : "RedshiftCluster" }
        },
        "ParameterGroupName" : {
            "Description" : "Name of parameter group",
            "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
        },
        "RedshiftClusterSubnetGroupName" : {
            "Description" : "Name of cluster subnet group",
            "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
        },
        "RedshiftClusterSecurityGroupName" : {
            "Description" : "Name of cluster security group",
            "Value" : { "Ref" : "SecurityGroup" }
        }
    }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  DatabaseName:
    Description: The name of the first database to be created when the cluster is
      created
    Type: String
    Default: dev
    AllowedPattern: "([a-z]|[0-9])+"
  ClusterType:
    Description: The type of cluster
    Type: String
    Default: single-node
    AllowedValues:
      - single-node
      - multi-node
  NumberOfNodes:
    Description: The number of compute nodes in the cluster. For multi-node clusters,
      the NumberOfNodes parameter must be greater than 1
    Type: Number
    Default: '1'
  NodeType:
    Description: The type of node to be provisioned
    Type: String
    Default: ds2.xlarge
    AllowedValues:
      - ds2.xlarge
      - ds2.8xlarge
      - dc1.large
      - dc1.8xlarge
  MasterUsername:
    Description: The user name that is associated with the master user account for
      the cluster that is being created
    Type: String
    Default: defaultuser
    AllowedPattern: "([a-z])([a-z]|[0-9])*"
  MasterUserPassword:
    Description: The password that is associated with the master user account for
      the cluster that is being created.
    Type: String
    NoEcho: 'true'
  InboundTraffic:
    Description: Allow inbound traffic to the cluster from this CIDR range.
```

```
Type: String
MinLength: '9'
MaxLength: '18'
Default: 0.0.0.0/0
AllowedPattern: "(\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}/(\d{1,2})"
ConstraintDescription: must be a valid CIDR range of the form x.x.x.x/x.
PortNumber:
  Description: The port number on which the cluster accepts incoming connections.
  Type: Number
  Default: '5439'
Conditions:
  IsMultiNodeCluster:
    Fn::Equals:
      - Ref: ClusterType
      - multi-node
Resources:
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    DependsOn: AttachGateway
    Properties:
      ClusterType:
        Ref: ClusterType
      NumberOfNodes:
        Fn::If:
          - IsMultiNodeCluster
          - Ref: NumberOfNodes
          - Ref: AWS::NoValue
      NodeType:
        Ref: NodeType
      DBName:
        Ref: DatabaseName
      MasterUsername:
        Ref: MasterUsername
      MasterUserPassword:
        Ref: MasterUserPassword
      ClusterParameterGroupName:
        Ref: RedshiftClusterParameterGroup
      VpcSecurityGroupIds:
        - Ref: SecurityGroup
      ClusterSubnetGroupName:
        Ref: RedshiftClusterSubnetGroup
      PubliclyAccessible: 'true'
      Port:
        Ref: PortNumber
```

```
RedshiftClusterParameterGroup:
  Type: AWS::Redshift::ClusterParameterGroup
  Properties:
    Description: Cluster parameter group
    ParameterGroupFamily: redshift-1.0
    Parameters:
      - ParameterName: enable_user_activity_logging
        ParameterValue: 'true'
RedshiftClusterSubnetGroup:
  Type: AWS::Redshift::ClusterSubnetGroup
  Properties:
    Description: Cluster subnet group
    SubnetIds:
      - Ref: PublicSubnet
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
PublicSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock: 10.0.0.0/24
    VpcId:
      Ref: VPC
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group
    SecurityGroupIngress:
      - CidrIp:
          Ref: InboundTraffic
        FromPort:
          Ref: PortNumber
        ToPort:
          Ref: PortNumber
        IpProtocol: tcp
    VpcId:
      Ref: VPC
myInternetGateway:
  Type: AWS::EC2::InternetGateway
AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId:
```

```
    Ref: VPC
  InternetGatewayId:
    Ref: myInternetGateway
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: VPC
PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId:
      Ref: PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId:
      Ref: myInternetGateway
PublicSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId:
      Ref: PublicSubnet
    RouteTableId:
      Ref: PublicRouteTable
Outputs:
  ClusterEndpoint:
    Description: Cluster endpoint
    Value: !Sub "${RedshiftCluster.Endpoint.Address}:${RedshiftCluster.Endpoint.Port}"
  ClusterName:
    Description: Name of cluster
    Value:
      Ref: RedshiftCluster
  ParameterGroupName:
    Description: Name of parameter group
    Value:
      Ref: RedshiftClusterParameterGroup
  RedshiftClusterSubnetGroupName:
    Description: Name of cluster subnet group
    Value:
      Ref: RedshiftClusterSubnetGroup
  RedshiftClusterSecurityGroupName:
    Description: Name of cluster security group
    Value:
```

Ref: SecurityGroup

Consultez aussi

[AWS::Redshift::Cluster](#)

Extraits de modèle Amazon RDS

Rubriques

- [Ressource d'instance de base de données Amazon RDS](#)
- [Ressource d'instance de base de données Oracle Amazon RDS](#)
- [Ressource du DBSecurity groupe Amazon RDS pour la gamme CIDR](#)
- [Groupe Amazon RDS avec un DBSecurity groupe de EC2 sécurité Amazon](#)
- [Plusieurs groupes de sécurité VPC](#)
- [Instance de base de données Amazon RDS dans un groupe de sécurité VPC](#)

Ressource d'instance de base de données Amazon RDS

Cet exemple montre une ressource d'instance de base de données Amazon RDS avec un mot de passe d'utilisateur principal géré. Pour plus d'informations, consultez la section [Gestion des mots de passe avec AWS Secrets Manager](#) dans le Guide de l'utilisateur Amazon RDS et la section [Gestion des mots de passe avec AWS Secrets Manager](#) dans le Guide de l'utilisateur Amazon Aurora.

Comme la propriété `EngineVersion` facultative n'est pas spécifiée, la version par défaut du moteur est utilisée pour cette instance de base de données. Pour plus de détails sur la version du moteur par défaut et les autres paramètres par défaut, voir [Créer DBInstance](#). La `DBSecurityGroups` propriété autorise l'accès réseau aux `AWS::RDS::DBSecurityGroup` ressources nommées `MyDbSecurityByEC2SecurityGroup` et `MyDbSecurityByCIDRIPGroup`. Pour en savoir plus, consultez [AWS::RDS::DBInstance](#). La ressource DB Instance possède également un attribut `DeletionPolicy` défini sur `Snapshot`. Avec l'`SnapshotDeletionPolicy` ensemble, il CloudFormation prendra un instantané de cette instance de base de données avant de la supprimer lors de la suppression de la pile.

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
```

```

    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "MasterUsername" : "MyName",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}

```

YAML

```

MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    MasterUsername: MyName
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
    DeletionPolicy: Snapshot

```

Ressource d'instance de base de données Oracle Amazon RDS

Cet exemple crée une ressource d'instance de base de données (DB) Oracle avec un mot de passe d'utilisateur principal géré. Pour plus d'informations, consultez la section [Gestion des mots de passe avec AWS Secrets Manager](#) dans le Guide de l'utilisateur Amazon RDS. L'exemple spécifie le Engine comme oracle-ee avec un modèle de licence de bring-your-own-license. Pour plus de détails sur les paramètres des instances de base de données Oracle, voir [Create DBInstance](#). La propriété DBSecurity Groups autorise l'accès réseau aux AWS::RDS::DBSecurityGroup ressources nommées MyDbSecurityBy EC2 SecurityGroup et. MyDbSecurityBy CIDRIPGroup Pour en savoir plus, consultez [AWS::RDS::DBInstance](#). La ressource DB Instance possède également un

attribut `DeletionPolicy` défini sur `Snapshot`. Avec l'`SnapshotDeletionPolicy`ensemble, il CloudFormation prendra un instantané de cette instance de base de données avant de la supprimer lors de la suppression de la pile.

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "oracle-ee",
    "LicenseModel" : "bring-your-own-license",
    "MasterUsername" : "master",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: oracle-ee
    LicenseModel: bring-your-own-license
    MasterUsername: master
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
    DeletionPolicy: Snapshot
```


Ressource du DBSecurity groupe Amazon RDS pour la gamme CIDR

Cet exemple montre une ressource Amazon RDS DBSecurityGroup avec une autorisation d'entrée pour la plage CIDR spécifiée au format ddd.ddd.ddd.ddd/dd. [Pour plus de détails, consultez AWS : :RDS : : DBSecurity Group and Ingress.](#)

JSON

```
"MyDbSecurityByCIDRIPGroup" : {
  "Type" : "AWS::RDS::DBSecurityGroup",
  "Properties" : {
    "GroupDescription" : "Ingress for CIDRIP",
    "DBSecurityGroupIngress" : {
      "CIDRIP" : "192.168.0.0/32"
    }
  }
}
```

YAML

```
MyDbSecurityByCIDRIPGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    GroupDescription: Ingress for CIDRIP
    DBSecurityGroupIngress:
      CIDRIP: "192.168.0.0/32"
```

Groupe Amazon RDS avec un DBSecurity groupe de EC2 sécurité Amazon

Cet exemple montre une ressource de [DBSecuritygroupe AWS : :RDS : : dotée](#) d'une autorisation d'entrée provenant d'un groupe de EC2 sécurité Amazon référencé par. MyEc2SecurityGroup

Pour ce faire, vous définissez un groupe EC2 de sécurité, puis vous utilisez la Ref fonction intrinsèque pour faire référence au groupe EC2 de sécurité au sein de votreDBSecurityGroup.

JSON

```
"DBInstance" : {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
```

```

        "DBName"           : { "Ref" : "DBName" },
        "Engine"           : "MySQL",
        "MasterUsername"   : { "Ref" : "DBUsername" },
        "DBInstanceClass"  : { "Ref" : "DBClass" },
        "DBSecurityGroups" : [ { "Ref" : "DBSecurityGroup" } ],
        "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
        "MasterUserPassword": { "Ref" : "DBPassword" }
    }
},

"DBSecurityGroup": {
    "Type": "AWS::RDS::DBSecurityGroup",
    "Properties": {
        "DBSecurityGroupIngress": {
            "EC2SecurityGroupName": {
                "Fn::GetAtt": ["WebServerSecurityGroup", "GroupName"]
            }
        },
        "GroupDescription" : "Frontend Access"
    }
},

"WebServerSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
        "SecurityGroupIngress" : [
            { "IpProtocol" : "tcp", "FromPort" : 80, "ToPort" : 80, "CidrIp" :
"0.0.0.0/0"},
            { "IpProtocol" : "tcp", "FromPort" : 22, "ToPort" : 22, "CidrIp" : "0.0.0.0/0"}
        ]
    }
}
}

```

YAML

Cet exemple est extrait de l'exemple complet suivant : [Drupal_Single_Instance_With_RDS.template](#)

```

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL

```

```

MasterUsername:
  Ref: DBUsername
DBInstanceClass:
  Ref: DBClass
DBSecurityGroups:
  - Ref: DBSecurityGroup
AllocatedStorage:
  Ref: DBAllocatedStorage
MasterUserPassword:
  Ref: DBPassword
DBSecurityGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    DBSecurityGroupIngress:
      EC2SecurityGroupName:
        Ref: WebServerSecurityGroup
      GroupDescription: Frontend Access
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0

```

Plusieurs groupes de sécurité VPC

Cet exemple montre une ressource de [DBSecuritygroupe AWS : :RDS : :](#) avec autorisation d'entrée pour plusieurs groupes de EC2 sécurité Amazon VPC dans [AWS : :RDS : :](#). DBSecurity GroupIngress

JSON

```

{
  "Resources" : {
    "DBInstance" : {
      "Type" : "AWS::RDS::DBInstance",

```

```

    "Properties" : {
      "AllocatedStorage" : "5",
      "DBInstanceClass" : "db.t2.small",
      "DBName" : { "Ref" : "MyDBName" },
      "DBSecurityGroups" : [ { "Ref" : "DbSecurityByEC2SecurityGroup" } ],
      "DBSubnetGroupName" : { "Ref" : "MyDBSubnetGroup" },
      "Engine" : "MySQL",
      "MasterUserPassword" : { "Ref" : "MyDBPassword" },
      "MasterUsername" : { "Ref" : "MyDBUsername" }
    },
    "DeletionPolicy" : "Snapshot"
  },
  "DbSecurityByEC2SecurityGroup" : {
    "Type" : "AWS::RDS::DBSecurityGroup",
    "Properties" : {
      "GroupDescription" : "Ingress for Amazon EC2 security group",
      "EC2VpcId" : { "Ref" : "MyVPC" },
      "DBSecurityGroupIngress" : [ {
        "EC2SecurityGroupId" : "sg-b0ff1111",
        "EC2SecurityGroupOwnerId" : "111122223333"
      }, {
        "EC2SecurityGroupId" : "sg-ffd72222",
        "EC2SecurityGroupOwnerId" : "111122223333"
      } ]
    }
  }
}
}
}

```

YAML

```

Resources:
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      AllocatedStorage: '5'
      DBInstanceClass: db.t2.small
      DBName:
        Ref: MyDBName
      DBSecurityGroups:
        - Ref: DbSecurityByEC2SecurityGroup
      DBSubnetGroupName:
        Ref: MyDBSubnetGroup

```

```

Engine: MySQL
MasterUserPassword:
  Ref: MyDBPassword
MasterUsername:
  Ref: MyDBUsername
DeletionPolicy: Snapshot
DbSecurityByEC2SecurityGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    GroupDescription: Ingress for Amazon EC2 security group
    EC2VpcId:
      Ref: MyVPC
    DBSecurityGroupIngress:
      - EC2SecurityGroupId: sg-b0ff1111
        EC2SecurityGroupOwnerId: '111122223333'
      - EC2SecurityGroupId: sg-ffd72222
        EC2SecurityGroupOwnerId: '111122223333'

```

Instance de base de données Amazon RDS dans un groupe de sécurité VPC

Cet exemple montre une instance de base de données Amazon RDS associée à un groupe de sécurité Amazon EC2 VPC.

JSON

```

{
  "DBEC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Open database for access",
      "SecurityGroupIngress": [{
        "IpProtocol": "tcp",
        "FromPort": 3306,
        "ToPort": 3306,
        "SourceSecurityGroupName": { "Ref": "WebServerSecurityGroup" }
      }]
    }
  },
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName": { "Ref": "DBName" },
      "Engine": "MySQL",

```

```

    "MultiAZ"      : { "Ref": "MultiAZDatabase" },
    "MasterUsername" : { "Ref" : "DBUser" },
    "DBInstanceClass" : { "Ref" : "DBClass" },
    "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
    "MasterUserPassword": { "Ref" : "DBPassword" },
    "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "DBEC2SecurityGroup", "GroupId" ] } ]
  }
}
}

```

YAML

```

DBEC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Open database for access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 3306
        ToPort: 3306
        SourceSecurityGroupName:
          Ref: WebServerSecurityGroup
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MultiAZ:
      Ref: MultiAZDatabase
    MasterUsername:
      Ref: DBUser
    DBInstanceClass:
      Ref: DBClass
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword
    VPCSecurityGroups:
      - !GetAtt DBEC2SecurityGroup.GroupId

```

Extraits de modèle Route 53

Rubriques

- [Jeu d'enregistrements de ressources Amazon Route 53 avec l'ID ou le nom de la zone hébergée](#)
- [Utilisation RecordSetGroup pour configurer des ensembles d'enregistrements de ressources pondérés](#)
- [Utilisation RecordSetGroup pour configurer un jeu d'enregistrements de ressources alias](#)
- [Enregistrement de ressource Alias défini pour une CloudFront distribution](#)

Jeu d'enregistrements de ressources Amazon Route 53 avec l'ID ou le nom de la zone hébergée

Lorsque vous créez un ensemble d'enregistrements de ressources Amazon Route 53, vous devez spécifier la zone hébergée dans laquelle vous souhaitez l'ajouter. CloudFormation propose deux méthodes pour spécifier une zone hébergée :

- Vous pouvez spécifier explicitement la zone hébergée à l'aide de la propriété `HostedZoneId`.
- Vous pouvez avoir CloudFormation trouvé la zone hébergée à l'aide de la `HostedZoneName` propriété. Si vous utilisez la `HostedZoneName` propriété et que plusieurs zones hébergées portent le même nom, CloudFormation cela ne crée pas la pile.

Ajouter en RecordSet utilisant HostedZoneId

Cet exemple ajoute un ensemble d'enregistrements de ressources Amazon Route 53 contenant un enregistrement SPF pour le nom de domaine `mysite.example.com` qui utilise la propriété `HostedZoneId` pour spécifier la zone hébergée.

JSON

```
"myDNSRecord" : {
  "Type" : "AWS::Route53::RecordSet",
  "Properties" :
  {
    "HostedZoneId" : "Z3DG6IL3SJCGPX",
    "Name" : "mysite.example.com.",
    "Type" : "SPF",
    "TTL" : "900",
    "ResourceRecords" : [ "\"v=spf1 ip4:192.168.0.1/16 -all\"" ]
  }
}
```

```
}  
}
```

YAML

```
myDNSRecord:  
  Type: AWS::Route53::RecordSet  
  Properties:  
    HostedZoneId: Z3DG6IL3SJCGPX  
    Name: mysite.example.com.  
    Type: SPF  
    TTL: '900'  
    ResourceRecords:  
      - '"v=spf1 ip4:192.168.0.1/16 -all"'
```

Ajouter en RecordSet utilisant HostedZoneName

Cet exemple ajoute un jeu d'enregistrements de ressources Amazon Route 53 pour le nom de domaine « mysite.example.com » à l'aide de la propriété HostedZoneName pour spécifier la zone hébergée.

JSON

```
"myDNSRecord2" : {  
    "Type" : "AWS::Route53::RecordSet",  
    "Properties" : {  
        "HostedZoneName" : "example.com.",  
        "Name" : "mysite.example.com.",  
        "Type" : "A",  
        "TTL" : "900",  
        "ResourceRecords" : [  
            "192.168.0.1",  
            "192.168.0.2"  
        ]  
    }  
}
```

YAML

```
myDNSRecord2:  
  Type: AWS::Route53::RecordSet  
  Properties:
```



```
HostedZoneName: example.com.
Name: mysite.example.com.
Type: A
TTL: '900'
ResourceRecords:
- 192.168.0.1
- 192.168.0.2
```

Utilisation RecordSetGroup pour configurer des ensembles d'enregistrements de ressources pondérés

Cet exemple utilise un [AWS::Route53::RecordSetgroupe](#) pour configurer deux enregistrements CNAME pour « exemple.com ». zone hébergée « exemple.com ». La propriété RecordSets contient les jeux d'enregistrements CNAME pour le nom DNS « mysite.example.com ». Chaque jeu d'enregistrements contient un identifiant (SetIdentifier) et une pondération (Weight). La proportion du trafic Internet acheminé vers les ressources est basée sur les calculs suivants :

- Frontend One: $140 / (140 + 60) = 140 / 200 = 70 \%$
- Frontend Two: $60 / (140 + 60) = 60 / 200 = 30 \%$

Pour de plus amples informations sur les jeux d'enregistrements de ressources pondérés, veuillez consulter [Routage pondéré](#) dans le Amazon Route 53 Manuel du développeur.

JSON

```
"myDNSOne" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Weighted RR for my frontends.",
    "RecordSets" : [
      {
        "Name" : "mysite.example.com.",
        "Type" : "CNAME",
        "TTL" : "900",
        "SetIdentifier" : "Frontend One",
        "Weight" : "140",
        "ResourceRecords" : ["example-ec2.amazonaws.com"]
      },
      {
        "Name" : "mysite.example.com.",
```

```

        "Type" : "CNAME",
        "TTL" : "900",
        "SetIdentifier" : "Frontend Two",
        "Weight" : "60",
        "ResourceRecords" : ["example-ec2-larger.amazonaws.com"]
    }
]
}
}

```

YAML

```

myDNSOne:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Weighted RR for my frontends.
    RecordSets:
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend One
        Weight: '140'
        ResourceRecords:
          - example-ec2.amazonaws.com
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend Two
        Weight: '60'
        ResourceRecords:
          - example-ec2-larger.amazonaws.com

```

Utilisation RecordSetGroup pour configurer un jeu d'enregistrements de ressources alias

Les exemples suivants utilisent un [AWS::Route53::RecordSetgroupe](#) pour configurer un jeu d'enregistrements de ressources d'alias nommé `example.com` qui achemine le trafic vers un équilibreur de charge ELB version 1 (classique) et un équilibreur de charge version 2 (application ou réseau). La [AliasTarget](#) propriété spécifie l'ID de zone hébergée et le nom DNS pour le myELB LoadBalancer en utilisant la fonction `GetAtt` intrinsèque. `GetAtt` récupère différentes propriétés

de la myELB ressource, selon que vous acheminez le trafic vers un équilibreur de charge version 1 ou version 2 :

- Équilibreur de charge Version 1 : CanonicalHostedZoneNameID et DNSName
- Équilibreur de charge Version 2 : CanonicalHostedZoneID et DNSName

Pour plus d'informations sur les jeux d'enregistrements de ressources d'alias, veuillez consulter [Choix entre des enregistrements avec ou sans alias](#) dans le Guide du développeur Amazon Route 53.

JSON pour équilibreur de charge Version 1

```
"myELB" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
},
"myDNS" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
    "RecordSets" : [
      {
        "Name" : "example.com.",
        "Type" : "A",
        "AliasTarget" : {
          "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
"CanonicalHostedZoneNameID"] },
          "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
        }
      }
    ]
  }
}
```

YAML pour équilibreur de charge Version 1

```
myELB:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
      - Name: example.com.
        Type: A
        AliasTarget:
          HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneNameID'
          DNSName: !GetAtt 'myELB.DNSName'
```

JSON pour équilibreur de charge Version 2

```
{
  "myELB" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
      "Subnets" : [
        { "Ref": "SubnetAZ1" },
        { "Ref": "SubnetAZ2" }
      ]
    }
  },
  "myDNS" : {
    "Type" : "AWS::Route53::RecordSetGroup",
    "Properties" : {
      "HostedZoneName" : "example.com.",
      "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
      "RecordSets" : [
        {
          "Name" : "example.com.",
          "Type" : "A",
```

```

        "AliasTarget" : {
            "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
"CanonicalHostedZoneID"] },
            "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
        }
    }
]
}
}

```

YAML pour équilibreur de charge Version 2

```

myELB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets:
      - Ref: SubnetAZ1
      - Ref: SubnetAZ2
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
      - Name: example.com.
        Type: A
        AliasTarget:
          HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneID'
          DNSName: !GetAtt 'myELB.DNSName'

```

Enregistrement de ressource Alias défini pour une CloudFront distribution

L'exemple suivant crée un enregistrement alias A qui pointe un nom de domaine personnalisé vers une CloudFront distribution existante. `myHostedZoneID` est supposé être soit une référence à une `AWS::Route53::HostedZone` ressource réelle dans le même modèle, soit un paramètre. `myCloudFrontDistribution` fait référence à une `AWS::CloudFront::Distribution` ressource dans le même modèle. L'enregistrement d'alias utilise l'ID de zone CloudFront hébergée standard (Z2FDTNDATAQYW2) et résout automatiquement le nom de domaine de la distribution à l'aide de `Fn::GetAtt`. Cette configuration permet au trafic Web d'être acheminé du domaine personnalisé vers la CloudFront distribution sans nécessiter d'adresse IP.

Note

Lorsque vous créez des ensembles d'enregistrements de ressources alias, vous devez spécifier Z2FDTNDATAQYW2 pour la propriété HostedZoneId. Les ensembles d'enregistrements de ressources d'alias pour ne CloudFront peuvent pas être créés dans une zone privée.

JSON

```
{
  "myDNS": {
    "Type": "AWS::Route53::RecordSetGroup",
    "Properties": {
      "HostedZoneId": {
        "Ref": "myHostedZoneID"
      },
      "RecordSets": [
        {
          "Name": {
            "Ref": "myRecordSetDomainName"
          },
          "Type": "A",
          "AliasTarget": {
            "HostedZoneId": "Z2FDTNDATAQYW2",
            "DNSName": {
              "Fn::GetAtt": [
                "myCloudFrontDistribution",
                "DomainName"
              ]
            },
            "EvaluateTargetHealth": false
          }
        }
      ]
    }
  }
}
```

YAML

```
myDNS:
```

```
Type: AWS::Route53::RecordSetGroup
Properties:
  HostedZoneId: !Ref myHostedZoneID
  RecordSets:
    - Name: !Ref myRecordSetDomainName
      Type: A
      AliasTarget:
        HostedZoneId: Z2FDTNDATAQYW2
        DNSName: !GetAtt
          - myCloudFrontDistribution
          - DomainName
      EvaluateTargetHealth: false
```

Extraits de modèle Amazon S3

Utilisez ces exemples de modèles Amazon S3 pour décrire vos compartiments Amazon S3 avec CloudFormation. Pour plus d'exemples, consultez la section [Exemples](#) dans la ressource `AWS::S3::Bucket`.

Rubriques

- [Création d'un compartiment Amazon S3 avec les paramètres par défaut](#)
- [Création d'un compartiment Amazon S3 pour l'hébergement d'un site web et avec une `DeletionPolicy`](#)
- [Création d'un site web statique à l'aide d'un domaine personnalisé](#)

Création d'un compartiment Amazon S3 avec les paramètres par défaut

Cet exemple utilise une propriété `AWS::S3::Bucket` pour créer un compartiment avec les paramètres par défaut.

JSON

```
"myS3Bucket" : {
  "Type" : "AWS::S3::Bucket"
}
```

YAML

```
MyS3Bucket:
```

Type: AWS::S3::Bucket

Création d'un compartiment Amazon S3 pour l'hébergement d'un site web et avec une **DeletionPolicy**

Cet exemple crée un compartiment en tant que site Web et désactive l'option Bloquer l'accès public (des autorisations de lecture publiques sont requises pour les compartiments configurés pour l'hébergement de sites Web). Une stratégie de compartiment public est ensuite ajoutée au compartiment. Comme cette ressource de compartiment possède un `DeletionPolicy` attribut défini sur `Retain`, elle ne CloudFormation supprimera pas ce compartiment lorsqu'elle supprimera la pile. La section `Output` sert à récupérer `Fn::GetAtt` pour extraire l'attribut `WebsiteURL` et l'attribut `DomainName` de la ressource `S3Bucket`.

Note

Les exemples suivants supposent que les paramètres `BlockPublicPolicy` et `RestrictPublicBuckets` blocage de l'accès public ont été désactivés au niveau du compte.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "S3Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "PublicAccessBlockConfiguration": {
          "BlockPublicAcls": false,
          "BlockPublicPolicy": false,
          "IgnorePublicAcls": false,
          "RestrictPublicBuckets": false
        },
        "WebsiteConfiguration": {
          "IndexDocument": "index.html",
          "ErrorDocument": "error.html"
        }
      },
      "DeletionPolicy": "Retain",
      "UpdateReplacePolicy": "Retain"
    }
  }
}
```



```

    },
    "BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "PolicyDocument": {
          "Id": "MyPolicy",
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "PublicReadForGetBucketObjects",
              "Effect": "Allow",
              "Principal": "*",
              "Action": "s3:GetObject",
              "Resource": {
                "Fn::Join": [
                  "",
                  [
                    "arn:aws:s3:::",
                    {
                      "Ref": "S3Bucket"
                    },
                    "/*"
                  ]
                ]
              }
            }
          ]
        },
        "Bucket": {
          "Ref": "S3Bucket"
        }
      }
    }
  },
  "Outputs": {
    "WebsiteURL": {
      "Value": {
        "Fn::GetAtt": [
          "S3Bucket",
          "WebsiteURL"
        ]
      },
      "Description": "URL for website hosted on S3"
    }
  },

```

```

    "S3BucketSecureURL": {
      "Value": {
        "Fn::Join": [
          "",
          [
            "https://",
            {
              "Fn::GetAtt": [
                "S3Bucket",
                "DomainName"
              ]
            }
          ]
        ],
        "Description": "Name of S3 bucket to hold website content"
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: error.html
      DeletionPolicy: Retain
      UpdateReplacePolicy: Retain
  BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      PolicyDocument:
        Id: MyPolicy
        Version: 2012-10-17

```

```

Statement:
  - Sid: PublicReadForGetBucketObjects
    Effect: Allow
    Principal: '*'
    Action: 's3:GetObject'
    Resource: !Join
      - ''
      - - 'arn:aws:s3:::'
        - !Ref S3Bucket
      - /*
    Bucket: !Ref S3Bucket
Outputs:
  WebsiteURL:
    Value: !GetAtt
      - S3Bucket
      - WebsiteURL
    Description: URL for website hosted on S3
  S3BucketSecureURL:
    Value: !Join
      - ''
      - - 'https://'
        - !GetAtt
          - S3Bucket
          - DomainName
    Description: Name of S3 bucket to hold website content

```

Création d'un site web statique à l'aide d'un domaine personnalisé

Vous pouvez utiliser Route 53 avec un domaine enregistré. L'exemple suivant part du principe que vous avez déjà créé une zone hébergée dans Route 53 pour votre domaine. Cet exemple crée deux compartiments pour l'hébergement du site web. Le compartiment racine héberge le contenu, tandis que l'autre renvoie les requêtes `www.domainname.com` vers le compartiment racine. Les jeux d'enregistrements mappent votre nom de domaine avec les points de terminaison Amazon S3.

Vous devrez également ajouter une politique de compartiment, comme indiqué dans les exemples ci-dessus.

Pour plus d'informations sur l'utilisation d'un domaine personnalisé, consultez le [Didacticiel : Configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré avec Route 53](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

Note

Les exemples suivants supposent que les paramètres `BlockPublicPolicy` et `RestrictPublicBuckets` blocage de l'accès public ont été désactivés au niveau du compte.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "S3hostedzoneID" : "Z3AQBSTGFYJSTF", "websiteendpoint" :
"s3-website-us-east-1.amazonaws.com" },
      "us-west-1" : { "S3hostedzoneID" : "Z2F56UZL2M1ACD", "websiteendpoint" :
"s3-website-us-west-1.amazonaws.com" },
      "us-west-2" : { "S3hostedzoneID" : "Z3BJ6K6RIION7M", "websiteendpoint" :
"s3-website-us-west-2.amazonaws.com" },
      "eu-west-1" : { "S3hostedzoneID" : "Z1BKCTXD74EZPE", "websiteendpoint" :
"s3-website-eu-west-1.amazonaws.com" },
      "ap-southeast-1" : { "S3hostedzoneID" : "Z300J2DXBE1FTB",
"websiteendpoint" : "s3-website-ap-southeast-1.amazonaws.com" },
      "ap-southeast-2" : { "S3hostedzoneID" : "Z1WCIGYICN2BYD",
"websiteendpoint" : "s3-website-ap-southeast-2.amazonaws.com" },
      "ap-northeast-1" : { "S3hostedzoneID" : "Z2M4EHUR26P7ZW",
"websiteendpoint" : "s3-website-ap-northeast-1.amazonaws.com" },
      "sa-east-1" : { "S3hostedzoneID" : "Z31GFT0UA1I2HV", "websiteendpoint" :
"s3-website-sa-east-1.amazonaws.com" }
    }
  },
  "Parameters": {
    "RootDomainName": {
      "Description": "Domain name for your website (example.com)",
      "Type": "String"
    }
  },
  "Resources": {
    "RootBucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "BucketName" : {"Ref":"RootDomainName"},

```

```

        "PublicAccessBlockConfiguration": {
            "BlockPublicAcls": false,
            "BlockPublicPolicy": false,
            "IgnorePublicAcls": false,
            "RestrictPublicBuckets": false
        },
        "WebsiteConfiguration": {
            "IndexDocument": "index.html",
            "ErrorDocument": "404.html"
        }
    },
    "WWWBucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "BucketName": {
                "Fn::Join": ["", ["www.", {"Ref": "RootDomainName"}]]
            },
            "AccessControl": "BucketOwnerFullControl",
            "WebsiteConfiguration": {
                "RedirectAllRequestsTo": {
                    "HostName": {"Ref": "RootBucket"}
                }
            }
        }
    },
    "myDNS": {
        "Type": "AWS::Route53::RecordSetGroup",
        "Properties": {
            "HostedZoneName": {
                "Fn::Join": ["", [{"Ref": "RootDomainName"}, "."]]
            },
            "Comment": "Zone apex alias.",
            "RecordSets": [
                {
                    "Name": {"Ref": "RootDomainName"},
                    "Type": "A",
                    "AliasTarget": {
                        "HostedZoneId": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "S3hostedzoneID"]},
                        "DNSName": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "websiteendpoint"]}
                    }
                }
            ]
        }
    },

```

```

        {
            "Name": {
                "Fn::Join": [ "", [ "www.", { "Ref": "RootDomainName" } ] ]
            },
            "Type": "CNAME",
            "TTL" : "900",
            "ResourceRecords" : [
                { "Fn::GetAtt": [ "WWWBucket", "DomainName" ] }
            ]
        }
    ]
}
},
"Outputs": {
    "WebsiteURL": {
        "Value": { "Fn::GetAtt": [ "RootBucket", "WebsiteURL" ] },
        "Description": "URL for website hosted on S3"
    }
}
}

```

YAML

```

Parameters:
  RootDomainName:
    Description: Domain name for your website (example.com)
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      S3hostedzoneID: Z3AQBSTGFYJSTF
      websiteendpoint: s3-website-us-east-1.amazonaws.com
    us-west-1:
      S3hostedzoneID: Z2F56UZL2M1ACD
      websiteendpoint: s3-website-us-west-1.amazonaws.com
    us-west-2:
      S3hostedzoneID: Z3BJ6K6RIION7M
      websiteendpoint: s3-website-us-west-2.amazonaws.com
    eu-west-1:
      S3hostedzoneID: Z1BKCTXD74EZPE
      websiteendpoint: s3-website-eu-west-1.amazonaws.com
    ap-southeast-1:

```

```

    S3hostedzoneID: Z300J2DXBE1FTB
    websiteendpoint: s3-website-ap-southeast-1.amazonaws.com
  ap-southeast-2:
    S3hostedzoneID: Z1WCIGYICN2BYD
    websiteendpoint: s3-website-ap-southeast-2.amazonaws.com
  ap-northeast-1:
    S3hostedzoneID: Z2M4EHUR26P7ZW
    websiteendpoint: s3-website-ap-northeast-1.amazonaws.com
  sa-east-1:
    S3hostedzoneID: Z31GFT0UA1I2HV
    websiteendpoint: s3-website-sa-east-1.amazonaws.com

```

Resources:

RootBucket:

Type: AWS::S3::Bucket

Properties:

```

    BucketName: !Ref RootDomainName
    PublicAccessBlockConfiguration:
      BlockPublicAcls: false
      BlockPublicPolicy: false
      IgnorePublicAcls: false
      RestrictPublicBuckets: false
    WebsiteConfiguration:
      IndexDocument: index.html
      ErrorDocument: 404.html

```

WWWBucket:

Type: AWS::S3::Bucket

Properties:

```

    BucketName: !Sub
      - www.${Domain}
      - Domain: !Ref RootDomainName
    AccessControl: BucketOwnerFullControl
    WebsiteConfiguration:
      RedirectAllRequestsTo:
        HostName: !Ref RootBucket

```

myDNS:

Type: AWS::Route53::RecordSetGroup

Properties:

```

    HostedZoneName: !Sub
      - ${Domain}.
      - Domain: !Ref RootDomainName
    Comment: Zone apex alias.
    RecordSets:
      - Name: !Ref RootDomainName
        Type: A

```

```

    AliasTarget:
      HostedZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', S3hostedzoneID]
      DNSName: !FindInMap [ RegionMap, !Ref 'AWS::Region', websiteendpoint]
    - Name: !Sub
      - www.${Domain}
      - Domain: !Ref RootDomainName
    Type: CNAME
    TTL: 900
    ResourceRecords:
      - !GetAtt WWWBucket.DomainName

Outputs:
  WebsiteURL:
    Value: !GetAtt RootBucket.WebsiteURL
    Description: URL for website hosted on S3

```

Extraits de modèle Amazon SNS

Cet exemple présente une ressource de rubrique Amazon SNS. Elle nécessite une adresse e-mail valide.

JSON

```

"MySNSTopic" : {
  "Type" : "AWS::SNS::Topic",
  "Properties" : {
    "Subscription" : [ {
      "Endpoint" : "add valid email address",
      "Protocol" : "email"
    } ]
  }
}

```

YAML

```

MySNSTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: "add valid email address"
      Protocol: email

```


Extraits de modèle Amazon SQS

Cet exemple présente une file d'attente Amazon SQS.

JSON

```
"MyQueue" : {  
  "Type" : "AWS::SQS::Queue",  
  "Properties" : {  
    "VisibilityTimeout" : "value"  
  }  
}
```

YAML

```
MyQueue:  
  Type: AWS::SQS::Queue  
  Properties:  
    VisibilityTimeout: value
```

Extraits de modèles Amazon Timestream

Amazon Timestream for InfluxDB permet aux développeurs d'applications et aux équipes d'exécuter facilement des bases AWS de données InfluxDB entièrement gérées pour DevOps des applications de séries chronologiques en temps réel utilisant l'open source. APIs Vous pouvez rapidement créer une base de données InfluxDB qui gère des charges de travail de séries chronologiques exigeantes. En quelques appels d'API simples, vous pouvez configurer, migrer, exploiter et dimensionner une base de données InfluxDB AWS grâce à des correctifs logiciels, à des sauvegardes et à une restauration automatisés. Vous pouvez également trouver ces exemples sur [awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb](https://github.com/aws-labs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb) on. GitHub

Rubriques

- [Exemple minimal utilisant les valeurs par défaut](#)
- [Exemple plus complet avec paramètres](#)

Ces CloudFormation modèles créent les ressources suivantes qui sont nécessaires pour créer, se connecter et surveiller avec succès une instance Amazon Timestream pour InfluxDB :

Amazon VPC

- VPC
- Un ou plusieurs Subnet
- InternetGateway
- RouteTable
- SecurityGroup

Amazon S3

- Bucket

Amazon Timestream

- InfluxDBInstance

Exemple minimal utilisant les valeurs par défaut

Cet exemple déploie une instance multi-AZ et accessible au public en utilisant les valeurs par défaut lorsque cela est possible.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBPassword"
          ]
        }
      ],
      "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
      }
    }
  }
}
```

```

},
"Parameters": {
  "DbInstanceName": {
    "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
    "Type": "String",
    "Default": "mydbinstance",
    "MinLength": 3,
    "MaxLength": 40,
    "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
  },
  "InfluxDBPassword": {
    "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,
    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  }
},
"Resources": {
  "VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {"CidrBlock": "10.0.0.0/16"}
  },
  "InternetGateway": {"Type": "AWS::EC2::InternetGateway"},
  "InternetGatewayAttachment": {
    "Type": "AWS::EC2::VPCGatewayAttachment",
    "Properties": {
      "InternetGatewayId": {"Ref": "InternetGateway"},
      "VpcId": {"Ref": "VPC"}
    }
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [

```

```

        0,
        {"Fn::GetAZs": ""}
    ]
},
"CidrBlock": {
    "Fn::Select": [
        0,
        {
            "Fn::Cidr": [
                {
                    "Fn::GetAtt": [
                        "VPC",
                        "CidrBlock"
                    ]
                },
                2,
                12
            ]
        }
    ]
},
"MapPublicIpOnLaunch": true
}
},
"Subnet2": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
        "VpcId": {"Ref": "VPC"},
        "AvailabilityZone": {
            "Fn::Select": [
                1,
                {"Fn::GetAZs": ""}
            ]
        },
        "CidrBlock": {
            "Fn::Select": [
                1,
                {
                    "Fn::Cidr": [
                        {
                            "Fn::GetAtt": [
                                "VPC",
                                "CidrBlock"
                            ]
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        },
        2,
        12
    ]
}
]
},
    "MapPublicIpOnLaunch": true
}
},
"RouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {"Ref": "VPC"}
    }
},
"DefaultRoute": {
    "Type": "AWS::EC2::Route",
    "DependsOn": "InternetGatewayAttachment",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
    }
},
"Subnet1RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
    }
},
"Subnet2RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet2"}
    }
},
"InfluxDBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupName": "influxdb-sg",
        "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",

```

```

    "VpcId": {"Ref": "VPC"}
  },
  "InfluxDBSecurityGroupIngress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupId": {"Ref": "InfluxDBSecurityGroup"},
      "IpProtocol": "tcp",
      "CidrIp": "0.0.0.0/0",
      "FromPort": 8086,
      "ToPort": 8086
    }
  },
  "InfluxDBLogsS3Bucket": {
    "Type": "AWS::S3::Bucket",
    "DeletionPolicy": "Retain"
  },
  "InfluxDBLogsS3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*"},
            "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
          },
          {
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": [
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
            ],
            "Principal": "*",
            "Condition": {
              "Bool": {"aws:SecureTransport": false}
            }
          }
        ]
      }
    }
  }
]

```

```

    }
  }
},
"DbInstance": {
  "Type": "AWS::Timestream::InfluxDBInstance",
  "DependsOn": "InfluxDBLogsS3BucketPolicy",
  "Properties": {
    "AllocatedStorage": 20,
    "DbInstanceType": "db.influx.medium",
    "Name": {"Ref": "DbInstanceName"},
    "Password": {"Ref": "InfluxDBPassword"},
    "PubliclyAccessible": true,
    "DeploymentType": "WITH_MULTIAZ_STANDBY",
    "VpcSecurityGroupIds": [
      {"Ref": "InfluxDBSecurityGroup"}
    ],
    "VpcSubnetIds": [
      {"Ref": "Subnet1"},
      {"Ref": "Subnet2"}
    ],
    "LogDeliveryConfiguration": {
      "S3Configuration": {
        "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
        "Enabled": true
      }
    }
  }
},
"Outputs": {
  "VPC": {
    "Description": "A reference to the VPC used to create network resources",
    "Value": {"Ref": "VPC"}
  },
  "Subnets": {
    "Description": "A list of the subnets created",
    "Value": {
      "Fn::Join": [
        ",",
        [
          {"Ref": "Subnet1"},
          {"Ref": "Subnet2"}
        ]
      ]
    }
  }
}

```

```

    }
  },
  "Subnet1": {
    "Description": "A reference to the subnet in the 1st Availability Zone",
    "Value": {"Ref": "Subnet1"}
  },
  "Subnet2": {
    "Description": "A reference to the subnet in the 2nd Availability Zone",
    "Value": {"Ref": "Subnet2"}
  },
  "InfluxDBSecurityGroup": {
    "Description": "Security group with port 8086 ingress rule",
    "Value": {"Ref": "InfluxDBSecurityGroup"}
  },
  "InfluxDBLogsS3Bucket": {
    "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
    "Value": {"Ref": "InfluxDBLogsS3Bucket"}
  },
  "DbInstance": {
    "Description": "A reference to the Timestream for InfluxDB DB instance",
    "Value": {"Ref": "DbInstance"}
  },
  "InfluxAuthParametersSecretArn": {
    "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
    "Value": {
      "Fn::GetAtt": [
        "DbInstance",
        "InfluxAuthParametersSecretArn"
      ]
    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",

```



```
}  
}  
}  
}  
]  
]  
],  
    "Endpoint"  
]
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBPassword
    ParameterLabels:
      VPCCIDR:
        default: VPC CIDR

Parameters:
  DbInstanceName:
    Description: The name that uniquely identifies the DB instance when interacting
with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a
prefix included in the endpoint. DB instance names must be unique per customer and per
Region.
    Type: String
    Default: mydbinstance
    MinLength: 3
    MaxLength: 40
    AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$
  InfluxDBPassword:
    Description: The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.
    Type: String
```

```
NoEcho: true
MinLength: 8
MaxLength: 64
AllowedPattern: ^[a-zA-Z0-9]+$
```

Resources:**VPC:**

```
Type: AWS::EC2::VPC
Properties:
  CidrBlock: 10.0.0.0/16
```

InternetGateway:

```
Type: AWS::EC2::InternetGateway
```

InternetGatewayAttachment:

```
Type: AWS::EC2::VPCElasticNetworkAttachment
Properties:
  InternetGatewayId: !Ref InternetGateway
  VpcId: !Ref VPC
```

Subnet1:

```
Type: AWS::EC2::Subnet
Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [0, !GetAZs '']
  CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
  MapPublicIpOnLaunch: true
```

Subnet2:

```
Type: AWS::EC2::Subnet
Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [1, !GetAZs '']
  CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
  MapPublicIpOnLaunch: true
```

RouteTable:

```
Type: AWS::EC2::RouteTable
Properties:
  VpcId: !Ref VPC
```

DefaultRoute:

```
Type: AWS::EC2::Route
DependsOn: InternetGatewayAttachment
Properties:
  RouteTableId: !Ref RouteTable
  DestinationCidrBlock: 0.0.0.0/0
  GatewayId: !Ref InternetGateway
```

Subnet1RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
```

```
Properties:
  RouteTableId: !Ref RouteTable
  SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !Ref InfluxDBSecurityGroup
    IpProtocol: tcp
    CidrIp: 0.0.0.0/0
    FromPort: 8086
    ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
          Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
          Principal:
            Service: timestream-influxdb.amazonaws.com
        - Action: "s3:*"
          Effect: Deny
          Resource:
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
          Principal: "*"
      Condition:
```

```

        Bool:
            aws:SecureTransport: false
    DbInstance:
        Type: AWS::Timestream::InfluxDBInstance
        DependsOn: InfluxDBLogsS3BucketPolicy
        Properties:
            AllocatedStorage: 20
            DbInstanceType: db.influx.medium
            Name: !Ref DbInstanceName
            Password: !Ref InfluxDBPassword
            PubliclyAccessible: true
            DeploymentType: WITH_MULTIAZ_STANDBY
            VpcSecurityGroupIds:
                - !Ref InfluxDBSecurityGroup
            VpcSubnetIds:
                - !Ref Subnet1
                - !Ref Subnet2
            LogDeliveryConfiguration:
                S3Configuration:
                    BucketName: !Ref InfluxDBLogsS3Bucket
                    Enabled: true

Outputs:
    # Network Resources
    VPC:
        Description: A reference to the VPC used to create network resources
        Value: !Ref VPC
    Subnets:
        Description: A list of the subnets created
        Value: !Join ["", [!Ref Subnet1, !Ref Subnet2]]
    Subnet1:
        Description: A reference to the subnet in the 1st Availability Zone
        Value: !Ref Subnet1
    Subnet2:
        Description: A reference to the subnet in the 2nd Availability Zone
        Value: !Ref Subnet2
    InfluxDBSecurityGroup:
        Description: Security group with port 8086 ingress rule
        Value: !Ref InfluxDBSecurityGroup

    # Timestream for InfluxDB Resources
    InfluxDBLogsS3Bucket:
        Description: S3 Bucket containing InfluxDB logs from the DB instance
        Value: !Ref InfluxDBLogsS3Bucket

```

```

DbInstance:
  Description: A reference to the Timestream for InfluxDB DB instance
  Value: !Ref DbInstance
InfluxAuthParametersSecretArn:
  Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password."
  Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
Endpoint:
  Description: The endpoint URL to connect to InfluxDB
  Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]

```

Exemple plus complet avec paramètres

Ce modèle d'exemple modifie dynamiquement les ressources réseau en fonction des paramètres fournis. Les paramètres incluent `PubliclyAccessible` et `DeploymentType`.

JSON

```

{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Network Configuration"},
          "Parameters": ["VPCCIDR"]
        },
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBUsername",
            "InfluxDBPassword",
            "InfluxDBOrganization",
            "InfluxDBBucket",
            "DbInstanceType",
            "DbStorageType",
            "AllocatedStorage",
            "PubliclyAccessible",
            "DeploymentType"
          ]
        }
      ]
    }
  }
}

```

```

    }
  ],
  "ParameterLabels": {
    "VPCCIDR": {"default": "VPC CIDR"}
  }
},
"Parameters": {
  "VPCCIDR": {
    "Description": "Please enter the IP range (CIDR notation) for the new VPC",
    "Type": "String",
    "Default": "10.0.0.0/16"
  },
  "DbInstanceName": {
    "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
    "Type": "String",
    "Default": "mydbinstance",
    "MinLength": 3,
    "MaxLength": 40,
    "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
  },
  "InfluxDBUsername": {
    "Description": "The username of the initial admin user created in InfluxDB.
Must start with a letter and can't end with a hyphen or contain two consecutive
hyphens. For example, my-user1. This username will allow you to access the InfluxDB
UI to perform various administrative tasks and also use the InfluxDB CLI to
create an operator token. These attributes will be stored in a Secret created in
AWS Secrets Manager in your account.",
    "Type": "String",
    "Default": "admin",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBPassword": {
    "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,

```

```

    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  },
  "InfluxDBOrganization": {
    "Description": "The name of the initial organization for the initial admin user
in InfluxDB. An InfluxDB organization is a workspace for a group of users.",
    "Type": "String",
    "Default": "org",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBBucket": {
    "Description": "The name of the initial InfluxDB bucket. All InfluxDB data
is stored in a bucket. A bucket combines the concept of a database and a retention
period (the duration of time that each data point persists). A bucket belongs to an
organization.",
    "Type": "String",
    "Default": "bucket",
    "MinLength": 2,
    "MaxLength": 64,
    "AllowedPattern": "^[^_\\\\""][^\\\\""]*$"
  },
  "DeploymentType": {
    "Description": "Specifies whether the Timestream for InfluxDB is deployed as
Single-AZ or with a MultiAZ Standby for High availability",
    "Type": "String",
    "Default": "WITH_MULTIAZ_STANDBY",
    "AllowedValues": [
      "SINGLE_AZ",
      "WITH_MULTIAZ_STANDBY"
    ]
  },
  "AllocatedStorage": {
    "Description": "The amount of storage to allocate for your DB storage type in GiB
(gibibytes).",
    "Type": "Number",
    "Default": 400,
    "MinValue": 20,
    "MaxValue": 16384
  },
  "DbInstanceType": {
    "Description": "The Timestream for InfluxDB DB instance type to run InfluxDB
on.",
    "Type": "String",

```

```

    "Default": "db.influx.medium",
    "AllowedValues": [
        "db.influx.medium",
        "db.influx.large",
        "db.influx.xlarge",
        "db.influx.2xlarge",
        "db.influx.4xlarge",
        "db.influx.8xlarge",
        "db.influx.12xlarge",
        "db.influx.16xlarge"
    ]
},
    "DbStorageType": {
        "Description": "The Timestream for InfluxDB DB storage type to read and write InfluxDB data.",
        "Type": "String",
        "Default": "InfluxIOIncludedT1",
        "AllowedValues": [
            "InfluxIOIncludedT1",
            "InfluxIOIncludedT2",
            "InfluxIOIncludedT3"
        ]
    },
    "PubliclyAccessible": {
        "Description": "Configures the DB instance with a public IP to facilitate access.",
        "Type": "String",
        "Default": true,
        "AllowedValues": [
            true,
            false
        ]
    }
},
    "Conditions": {
        "IsMultiAZ": {
            "Fn::Equals": [
                {"Ref": "DeploymentType"},
                "WITH_MULTIAZ_STANDBY"
            ]
        }
    },
    "IsPublic": {
        "Fn::Equals": [
            {"Ref": "PubliclyAccessible"},

```



```

        true
    ]
}
},
"Resources": {
    "VPC": {
        "Type": "AWS::EC2::VPC",
        "Properties": {
            "CidrBlock": {"Ref": "VPCCIDR"}
        }
    },
    "InternetGateway": {
        "Type": "AWS::EC2::InternetGateway",
        "Condition": "IsPublic"
    },
    "InternetGatewayAttachment": {
        "Type": "AWS::EC2::VPCGatewayAttachment",
        "Condition": "IsPublic",
        "Properties": {
            "InternetGatewayId": {"Ref": "InternetGateway"},
            "VpcId": {"Ref": "VPC"}
        }
    },
    "Subnet1": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "VpcId": {"Ref": "VPC"},
            "AvailabilityZone": {
                "Fn::Select": [
                    0,
                    {"Fn::GetAZs": ""}
                ]
            },
            "CidrBlock": {
                "Fn::Select": [
                    0,
                    {
                        "Fn::Cidr": [
                            {
                                "Fn::GetAtt": [
                                    "VPC",
                                    "CidrBlock"
                                ]
                            }
                        ]
                    }
                ]
            }
        }
    },

```

```

        2,
        12
    ]
}
]
},
"MapPublicIpOnLaunch": {
    "Fn::If": [
        "IsPublic",
        true,
        false
    ]
}
},
"Subnet2": {
    "Type": "AWS::EC2::Subnet",
    "Condition": "IsMultiAZ",
    "Properties": {
        "VpcId": {"Ref": "VPC"},
        "AvailabilityZone": {
            "Fn::Select": [
                1,
                {"Fn::GetAZs": ""}
            ]
        },
        "CidrBlock": {
            "Fn::Select": [
                1,
                {
                    "Fn::Cidr": [
                        {
                            "Fn::GetAtt": [
                                "VPC",
                                "CidrBlock"
                            ]
                        },
                        2,
                        12
                    ]
                }
            ]
        },
        "MapPublicIpOnLaunch": {

```

```

        "Fn::If": [
            "IsPublic",
            true,
            false
        ]
    }
},
"RouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {"Ref": "VPC"}
    }
},
"DefaultRoute": {
    "Type": "AWS::EC2::Route",
    "Condition": "IsPublic",
    "DependsOn": "InternetGatewayAttachment",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
    }
},
"Subnet1RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
    }
},
"Subnet2RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Condition": "IsMultiAZ",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet2"}
    }
},
"InfluxDBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupName": "influxdb-sg",
        "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",

```

```

    "VpcId": {"Ref": "VPC"}
  },
  "InfluxDBSecurityGroupIngress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupId": {"Ref": "InfluxDBSecurityGroup"},
      "IpProtocol": "tcp",
      "CidrIp": "0.0.0.0/0",
      "FromPort": 8086,
      "ToPort": 8086
    }
  },
  "InfluxDBLogsS3Bucket": {
    "Type": "AWS::S3::Bucket",
    "DeletionPolicy": "Retain"
  },
  "InfluxDBLogsS3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*"},
            "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
          },
          {
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": [
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
            ],
            "Principal": "*",
            "Condition": {
              "Bool": {"aws:SecureTransport": false}
            }
          }
        ]
      }
    }
  }
]

```

```

    }
  }
},
"DbInstance": {
  "Type": "AWS::Timestream::InfluxDBInstance",
  "DependsOn": "InfluxDBLogsS3BucketPolicy",
  "Properties": {
    "DbStorageType": {"Ref": "DbStorageType"},
    "AllocatedStorage": {"Ref": "AllocatedStorage"},
    "DbInstanceType": {"Ref": "DbInstanceType"},
    "Name": {"Ref": "DbInstanceName"},
    "Username": {"Ref": "InfluxDBUsername"},
    "Password": {"Ref": "InfluxDBPassword"},
    "Organization": {"Ref": "InfluxDBOrganization"},
    "Bucket": {"Ref": "InfluxDBBucket"},
    "PubliclyAccessible": {
      "Fn::If": [
        "IsPublic",
        true,
        false
      ]
    },
  },
  "DeploymentType": {"Ref": "DeploymentType"},
  "VpcSecurityGroupIds": [
    {"Ref": "InfluxDBSecurityGroup"}
  ],
  "VpcSubnetIds": {
    "Fn::If": [
      "IsMultiAZ",
      [
        {"Ref": "Subnet1"},
        {"Ref": "Subnet2"}
      ],
      [
        {"Ref": "Subnet1"}
      ]
    ]
  },
},
"LogDeliveryConfiguration": {
  "S3Configuration": {
    "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
    "Enabled": true
  }
}
}

```

```

    }
  },
  "Outputs": {
    "VPC": {
      "Description": "A reference to the VPC used to create network resources",
      "Value": {"Ref": "VPC"}
    },
    "Subnets": {
      "Description": "A list of the subnets created",
      "Value": {
        "Fn::If": [
          "IsMultiAZ",
          {
            "Fn::Join": [
              ",",
              [
                {"Ref": "Subnet1"},
                {"Ref": "Subnet2"}
              ]
            ]
          },
          {"Ref": "Subnet1"}
        ]
      }
    },
    "Subnet1": {
      "Description": "A reference to the subnet in the 1st Availability Zone",
      "Value": {"Ref": "Subnet1"}
    },
    "Subnet2": {
      "Condition": "IsMultiAZ",
      "Description": "A reference to the subnet in the 2nd Availability Zone",
      "Value": {"Ref": "Subnet2"}
    },
    "InfluxDBSecurityGroup": {
      "Description": "Security group with port 8086 ingress rule",
      "Value": {"Ref": "InfluxDBSecurityGroup"}
    },
    "InfluxDBLogsS3Bucket": {
      "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
      "Value": {"Ref": "InfluxDBLogsS3Bucket"}
    },
    "DbInstance": {

```

```

    "Description": "A reference to the Timestream for InfluxDB DB instance",
    "Value": {"Ref": "DbInstance"}
  },
  "InfluxAuthParametersSecretArn": {
    "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
    "Value": {
      "Fn::GetAtt": [
        "DbInstance",
        "InfluxAuthParametersSecretArn"
      ]
    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",
              "Endpoint"
            ]
          },
          ":8086"
        ]
      ]
    }
  }
}

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:

```

```

    default: "Network Configuration"
Parameters:
  - VPCCIDR
-
Label:
  default: "Amazon Timestream for InfluxDB Configuration"
Parameters:
  - DbInstanceName
  - InfluxDBUsername
  - InfluxDBPassword
  - InfluxDBOrganization
  - InfluxDBBucket
  - DbInstanceType
  - DbStorageType
  - AllocatedStorage
  - PubliclyAccessible
  - DeploymentType
ParameterLabels:
  VPCCIDR:
    default: VPC CIDR

```

Parameters:

Network Configuration

VPCCIDR:

Description: Please enter the IP range (CIDR notation) for the new VPC

Type: String

Default: 10.0.0.0/16

Timestream for InfluxDB Configuration

DbInstanceName:

Description: The name that uniquely identifies the DB instance when interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a prefix included in the endpoint. DB instance names must be unique per customer and per Region.

Type: String

Default: mydbinstance

MinLength: 3

MaxLength: 40

AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*\$

InfluxDB initial user configurations

InfluxDBUsername:

Description: The username of the initial admin user created in InfluxDB. Must start with a letter and can't end with a hyphen or contain two consecutive hyphens. For example, my-user1. This username will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator

token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

Type: String
Default: admin
MinLength: 1
MaxLength: 64

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS in your account.

Type: String
NoEcho: true
MinLength: 8
MaxLength: 64
AllowedPattern: ^[a-zA-Z0-9]+\$

InfluxDBOrganization:

Description: The name of the initial organization for the initial admin user in InfluxDB. An InfluxDB organization is a workspace for a group of users.

Type: String
Default: org
MinLength: 1
MaxLength: 64

InfluxDBBucket:

Description: The name of the initial InfluxDB bucket. All InfluxDB data is stored in a bucket. A bucket combines the concept of a database and a retention period (the duration of time that each data point persists). A bucket belongs to an organization.

Type: String
Default: bucket
MinLength: 2
MaxLength: 64
AllowedPattern: ^[^\"]*[^\"]*\$

DeploymentType:

Description: Specifies whether the Timestream for InfluxDB is deployed as Single-AZ or with a MultiAZ Standby for High availability

Type: String
Default: WITH_MULTIAZ_STANDBY
AllowedValues:
- SINGLE_AZ
- WITH_MULTIAZ_STANDBY

AllocatedStorage:

Description: The amount of storage to allocate for your DB storage type in GiB (gibibytes).

Type: Number

Default: 400
MinValue: 20
MaxValue: 16384

DbInstanceType:

Description: The Timestream for InfluxDB DB instance type to run InfluxDB on.

Type: String

Default: db.influx.medium

AllowedValues:

- db.influx.medium
- db.influx.large
- db.influx.xlarge
- db.influx.2xlarge
- db.influx.4xlarge
- db.influx.8xlarge
- db.influx.12xlarge
- db.influx.16xlarge

DbStorageType:

Description: The Timestream for InfluxDB DB storage type to read and write InfluxDB data.

Type: String

Default: InfluxIOIncludedT1

AllowedValues:

- InfluxIOIncludedT1
- InfluxIOIncludedT2
- InfluxIOIncludedT3

PubliclyAccessible:

Description: Configures the DB instance with a public IP to facilitate access.

Type: String

Default: true

AllowedValues:

- true
- false

Conditions:

IsMultiAZ: !Equals [!Ref DeploymentType, WITH_MULTIAZ_STANDBY]

IsPublic: !Equals [!Ref PubliclyAccessible, true]

Resources:**VPC:**

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VPCCIDR

InternetGateway:

Type: AWS::EC2::InternetGateway

```
Condition: IsPublic
InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Condition: IsPublic
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC
Subnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
    MapPublicIpOnLaunch: !If [IsPublic, true, false]
Subnet2:
  Type: AWS::EC2::Subnet
  Condition: IsMultiAZ
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
    MapPublicIpOnLaunch: !If [IsPublic, true, false]
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
DefaultRoute:
  Type: AWS::EC2::Route
  Condition: IsPublic
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
Subnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Condition: IsMultiAZ
  Properties:
    RouteTableId: !Ref RouteTable
```

```
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !Ref InfluxDBSecurityGroup
    IpProtocol: tcp
    CidrIp: 0.0.0.0/0
    FromPort: 8086
    ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
          Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
          Principal:
            Service: timestream-influxdb.amazonaws.com
        - Action: "s3:*"
          Effect: Deny
          Resource:
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
            - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
          Principal: "*"
          Condition:
            Bool:
              aws:SecureTransport: false
DbInstance:
  Type: AWS::Timestream::InfluxDBInstance
  DependsOn: InfluxDBLogsS3BucketPolicy
  Properties:
    DbStorageType: !Ref DbStorageType
```

```

AllocatedStorage: !Ref AllocatedStorage
DbInstanceType: !Ref DbInstanceType
Name: !Ref DbInstanceName
Username: !Ref InfluxDBUsername
Password: !Ref InfluxDBPassword
Organization: !Ref InfluxDBOrganization
Bucket: !Ref InfluxDBBucket
PubliclyAccessible: !If [IsPublic, true, false]
DeploymentType: !Ref DeploymentType
VpcSecurityGroupIds:
  - !Ref InfluxDBSecurityGroup
VpcSubnetIds: !If
  - IsMultiAZ
  -
    - !Ref Subnet1
    - !Ref Subnet2
  -
    - !Ref Subnet1
LogDeliveryConfiguration:
  S3Configuration:
    BucketName: !Ref InfluxDBLogsS3Bucket
    Enabled: true

```

Outputs:

Network Resources

VPC:

Description: A reference to the VPC used to create network resources
 Value: !Ref VPC

Subnets:

Description: A list of the subnets created
 Value: !If
 - IsMultiAZ
 - !Join [",", [!Ref Subnet1, !Ref Subnet2]]
 - !Ref Subnet1

Subnet1:

Description: A reference to the subnet in the 1st Availability Zone
 Value: !Ref Subnet1

Subnet2:

Condition: IsMultiAZ
 Description: A reference to the subnet in the 2nd Availability Zone
 Value: !Ref Subnet2

InfluxDBSecurityGroup:

Description: Security group with port 8086 ingress rule
 Value: !Ref InfluxDBSecurityGroup

```
# Timestream for InfluxDB Resources
InfluxDBLogsS3Bucket:
  Description: S3 Bucket containing InfluxDB logs from the DB instance
  Value: !Ref InfluxDBLogsS3Bucket
DbInstance:
  Description: A reference to the Timestream for InfluxDB DB instance
  Value: !Ref DbInstance
InfluxAuthParametersSecretArn:
  Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
  containing the initial InfluxDB authorization parameters. The secret value is a JSON
  formatted key-value pair holding InfluxDB authorization values: organization, bucket,
  username, and password."
  Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
Endpoint:
  Description: The endpoint URL to connect to InfluxDB
  Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]
```

Déployez Windows des piles basées sur CloudFormation

Cette page fournit des liens vers la documentation de référence technique pour les CloudFormation ressources couramment utilisées dans les déploiements Windows basés.

CloudFormation fournit un support pour le déploiement et la gestion des Microsoft Windows piles via Infrastructure as Code (IaC). Vous pouvez l'utiliser CloudFormation pour le provisionnement automatisé d'instances EC2 Windows basées, sur SQL Server Amazon RDS, etc. Microsoft Active Directory Directory Service

AWS fournit des images Amazon Machine (AMIs) préconfigurées spécialement conçues pour les Windows plateformes afin de vous aider à déployer rapidement des applications sur Amazon EC2. Il s'agit AMIs notamment des Microsoft paramètres par défaut et AWS des personnalisations spécifiques. Avec CloudFormation, vous pouvez choisir une AMI appropriée, lancer une instance et y accéder via Remote Desktop Connection, comme vous le feriez avec n'importe quelle autre Windows Server. Ils AMIs contiennent des composants logiciels essentiels, notamment EC2Launch (les versions varient selon les Windows Server éditions) AWS Systems Manager, CloudFormation, Outils AWS pour PowerShell, et divers pilotes réseau, de stockage et graphiques pour garantir des performances optimales et une compatibilité avec les AWS services. Pour plus d'informations, consultez la [référence AMI AWS Windows](#).

CloudFormation prend également en charge les outils de configuration logicielle, tels que les UserData scripts, qui peuvent être exécutés PowerShell ou les commandes par lots lors du premier démarrage d'une instance EC2. Il propose aussi des scripts d'assistance (cfn-init, cfn-signal, cfn-get-metadata et cfn-hup) et prend en charge les métadonnées `AWS::CloudFormation::Init` pour la gestion des progiciels, des fichiers et des services sur les instances Windows.

Pour les environnements d'entreprise, CloudFormation permet l'adhésion à des domaines, la gestion des Windows licences via des modèles de licence EC2 et le traitement sécurisé des informations d'identification avec AWS Secrets Manager Associé à des modèles contrôlés par version et à des déploiements reproductibles, il CloudFormation aide les entreprises à maintenir des Windows environnements cohérents, sécurisés et évolutifs sur plusieurs comptes et comptes. Régions AWS

Pour plus de détails sur les CloudFormation ressources couramment utilisées dans les déploiements Windows basés, consultez les rubriques de référence techniques suivantes.

Type de ressource	Description
AWS::EC2::Instance	Pour le lancement des instances EC2 Windows.
AWS::EC2::SecurityGroup	Pour la définition des règles de pare-feu pour les charges de travail Windows.
AWS::AutoScaling::AutoScalingGroupe	Pour la mise à l'échelle des instances EC2 Windows.
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftANNONCE	Pour le déploiement de Microsoft Active Directory.
AWS::FSx::FileSystem	Pour le déploiement FSx pour Windows le serveur de fichiers.
AWS::RDS::DBInstance	Pour le provisionnement de SQL Server sur Amazon RDS.
AWS::CloudFormation::Init	Utilisé dans les métadonnées EC2 pour la configuration des instances. Pour de plus amples informations, veuillez consulter Stacks basés sur le bootstrap Windows CloudFormation .

Type de ressource	Description
AWS::SecretsManager::Secret	Pour la gestion sécurisée des informations d'identification et mots de passe Windows.
AWS::SSM::Parameter	Pour le stockage sécurisé des valeurs de configuration.
AWS::IAM::InstanceProfile AWS::IAM::Role	Pour l'attribution d'autorisations aux applications exécutées sur des instances EC2.

Stacks basés sur le bootstrap Windows CloudFormation

Cette rubrique décrit comment amorcer une pile Windows et comment résoudre les problèmes de création de la pile.

Rubriques

- [Données utilisateur dans les instances EC2](#)
- [CloudFormation scripts d'assistance](#)
- [Exemple d'amorçage d'une pile Windows](#)
- [Échappement des barres obliques inversées dans les chemins de fichiers Windows](#)
- [Gestion des services Windows](#)
- [Résolution des problèmes de création de pile](#)

Données utilisateur dans les instances EC2

Les données utilisateur sont une caractéristique Amazon EC2 qui vous permet de transmettre des scripts ou des informations de configuration à une instance EC2 lors de son lancement.

Pour les instances EC2 Windows :

- Vous pouvez utiliser soit des scripts batch (en utilisant des balises `<script>`), soit des scripts PowerShell (en utilisant des balises `<powershell>`).
- L'exécution du script est gérée par EC2Launch.

⚠ Important

Si vous créez votre propre Windows AMI à utiliser avec CloudFormation, assurez-vous que la EC2Launch version v2 est correctement configurée. EC2LaunchLa version v2 est requise pour que les outils d' CloudFormation amorçage initialisent et configurent correctement les Windows instances lors de la création de la pile. Pour plus d'informations, consultez [Utilisation de l'agent EC2Launch v2 pour exécuter des tâches lors du lancement d'une instance Windows EC2](#) dans le Guide de l'utilisateur Amazon EC2.

Pour plus d'informations à ce sujet AWS Windows AMIs, consultez la [référence de l'AWSWindowsAMI](#).

CloudFormation scripts d'assistance

Les scripts d'assistance sont des utilitaires permettant de configurer des instances durant le processus d'amorçage. Utilisés avec les données utilisateur Amazon EC2, ils offrent de puissantes capacités de configuration.

CloudFormation fournit les scripts d'assistance Python suivants que vous pouvez utiliser pour installer des logiciels et démarrer des services sur une instance Amazon EC2 que vous créez dans le cadre de votre stack :

- `cfn-init` : à utiliser pour extraire et interpréter les métadonnées des ressources, installer des packages, créer des fichiers et démarrer des services.
- `cfn-signal` : à utiliser pour envoyer un signal contenant une `CreationPolicy`, afin de synchroniser d'autres ressources dans la pile lorsque la ressource ou l'application préalable est prête.
- `cfn-get-metadata` : à utiliser pour extraire des métadonnées pour une ressource ou un chemin d'accès vers une clé spécifique.
- `cfn-hup` : à utiliser pour rechercher les mises à jour de métadonnées et exécuter des hooks personnalisés lorsque des modifications sont détectées.

Vous appelez les scripts directement à partir de votre modèle. Ces scripts interagissent avec les métadonnées des ressources qui sont définies dans le même modèle. Les scripts s'exécutent dans l'instance Amazon EC2 au cours du processus de création de la pile.

Pour plus d'informations, consultez la [référence aux scripts CloudFormation d'assistance](#) dans le Guide de référence des CloudFormation modèles.

Exemple d'amorçage d'une pile Windows

Examinons des exemples d'extraits provenant d'un modèle Windows Server qui effectue les actions suivantes :

- Lance une instance EC2 nommée `TestInstance` à partir d'une AMI Windows Server 2022.
- Crée un fichier de test simple pour vérifier que `cfn-init` fonctionne.
- Configure `cfn-hup` pour la gestion continue de la configuration.
- Utilise une `CreationPolicy` pour s'assurer que l'instance indique une exécution réussie.

Le script d'assistance `cfn-init` est utilisé pour chacune de ces actions, selon les données définies dans la ressource `AWS::CloudFormation::Init` du modèle.

La section `AWS::CloudFormation::Init` est nommée `TestInstance` et commence par la déclaration suivante.

```
TestInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        default:
          - create_files
          - start_services
```

Après cela, la section `files` de `AWS::CloudFormation::Init` est déclarée.

```
create_files:
  files:
    c:\cfn\test.txt:
      content: !Sub |
        Hello from ${AWS::StackName}
    c:\cfn\cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackName}
        region=${AWS::Region}
```

```
interval=2
c:\cfn\hooks.d\cfn-auto-reloader.conf:
content: !Sub |
    [cfn-auto-reloader-hook]
    triggers=post.update
    path=Resources.TestInstance.Metadata.AWS::CloudFormation::Init
    action=cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --
region ${AWS::Region}
```

Trois fichiers sont créés ici et placés dans le répertoire C:\cfn au niveau de l'instance de serveur :

- `test.txt`, un simple fichier de test qui vérifie que `cfn-init` fonctionne correctement et peut créer des fichiers contenant du contenu dynamique.
- `cfn-hup.conf`, le fichier de configuration pour `cfn-hup` avec un intervalle de vérification de 2 minutes.
- `cfn-auto-reloader.conf`, le fichier de configuration pour le hook utilisé par `cfn-hup` pour lancer une mise à jour (appelant `cfn-init`) lorsque les métadonnées dans `AWS::CloudFormation::Init` changent.

Vient ensuite la section `start_services`, qui configure les services Windows.

```
start_services:
  services:
    windows:
      cfn-hup:
        enabled: true
        ensureRunning: true
        files:
          - c:\cfn\cfn-hup.conf
          - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Cette section veille à ce que le service `cfn-hup` soit démarré et redémarre automatiquement si les fichiers de configuration sont modifiés. Le service surveille les modifications apportées aux CloudFormation métadonnées et s'exécute à nouveau `cfn-init` lorsque des mises à jour sont détectées.

Vient ensuite la section `Properties`.

```
TestInstance:
  Type: AWS::EC2::Instance
```

```
CreationPolicy:
  ResourceSignal:
    Timeout: PT20M
Metadata:
  AWS::CloudFormation::Init:
    # ... metadata configuration ...
Properties:
  InstanceType: t2.large
  ImageId: '{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base}}'
  SecurityGroupIds:
    - !Ref InstanceSecurityGroup
  KeyName: !Ref KeyPairName
  UserData:
    Fn::Base64: !Sub |
      <powershell>
        cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --region
${AWS::Region}
        cfn-signal.exe -e $lastexitcode --stack ${AWS::StackName} --resource
TestInstance --region ${AWS::Region}
      </powershell>
```

Dans cette section, la propriété `UserData` contient un script PowerShell qui sera exécuté par `EC2Launch`, entouré de balises `<powershell>`. Le script s'exécute `cfn-init` avec le `default` `ConfigSet`, puis l'utilise `cfn-signal` pour renvoyer le code de sortie à CloudFormation. La `CreationPolicy` est utilisée pour s'assurer que l'instance est correctement configurée avant que la création de la pile ne soit considérée comme terminée.

La propriété `ImageId` utilise un paramètre public du magasin de paramètres Systems Manager pour récupérer automatiquement le dernier ID d'AMI Windows Server 2022. Cette approche élimine le besoin de mappages d'AMI propres à chaque région et permet d'utiliser systématiquement la version la plus récente. L'utilisation des paramètres de Systems Manager pour l'AMI IDs est une bonne pratique pour conserver les références AMI actuelles. Si vous prévoyez de vous connecter à votre instance, assurez-vous que la propriété `SecurityGroupIds` référence un groupe de sécurité permettant l'accès RDP.

La `CreationPolicy` est déclarée comme partie des propriétés de la ressource et définit une période d'expiration. La commande `cfn-signal` contenue dans les données utilisateur indique que la configuration de l'instance est terminée :

```
TestInstance:
```

```
Type: AWS::EC2::Instance
CreationPolicy:
  ResourceSignal:
    Timeout: PT20M
Properties:
  # ... other properties ...
```

Comme le processus d'amorçage est minimal et ne crée que des fichiers et démarre des services, le délai d'CreationPolicyattente est de 20 minutes (PT20M) avant d'expirer. Le délai d'expiration est spécifié au format ISO 8601. Notez que les instances Windows mettent généralement plus de temps à se lancer que les instances Linux, donc testez soigneusement afin de déterminer la meilleure valeur d'expiration selon vos besoins.

Si tout se déroule correctement, la CreationPolicy se termine avec succès et vous pouvez accéder à l'instance Windows Server en utilisant son adresse IP publique. Une fois la création de la pile terminée, l'ID de l'instance et l'adresse IP publique seront affichés dans l'onglet Sorties de la CloudFormation console.

```
Outputs:
  InstanceId:
    Value: !Ref TestInstance
    Description: Instance ID of the Windows Server
  PublicIP:
    Value: !GetAtt TestInstance.PublicIp
    Description: Public IP address of the Windows Server
```

Vous pouvez également vérifier manuellement que le processus d'amorçage a fonctionné correctement en vous connectant à l'instance via RDP et en confirmant que le fichier `C:\cfn\test.txt` existe et contient le contenu attendu. Pour plus d'informations sur la connexion aux instances Windows, consultez [Connexion de votre instance Windows à l'aide de RDP](#) dans le Guide de l'utilisateur Amazon EC2.

Échappement des barres obliques inversées dans les chemins de fichiers Windows

Lorsque vous référencez Windows des chemins dans CloudFormation des modèles, pensez toujours à éviter correctement les barres obliques inverses (`\`) en fonction du format du modèle que vous utilisez.

- Dans les modèles JSON, vous devez utiliser des doubles barres obliques inversées dans les chemins de fichiers Windows, car JSON traite la barre oblique inversée comme un caractère

d'échappement. La première barre oblique inversée échappe la seconde, ce qui produit une seule barre interprétée littéralement.

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\\\SharePoint\\\\SharePointFoundation2010.exe /extract:C:\\\\SharePoint
\\\\SPF2010 /quiet /log:C:\\\\SharePoint\\\\SharePointFoundation2010-extract.log"
  }
}
```

- Pour les modèles YAML, des barres obliques inversées simples sont généralement suffisantes.

```
commands:
  1-extract:
    command: C:\SharePoint\SharePointFoundation2010.exe /extract:C:\SharePoint
\SPF2010 /quiet /log:C:\SharePoint\SharePointFoundation2010-extract.log
```

Gestion des services Windows

Vous gérez les services Windows de la même manière que les services Linux, sauf que vous utilisez la clé `windows` au lieu de `sysvinit`. L'exemple suivant démarre le service `cfn-hup`, définit son démarrage sur Automatique, et redémarre le service si `cfn-init` modifie les fichiers de configuration `c:\cfn\cfn-hup.conf` ou `c:\cfn\hooks.d\cfn-auto-reloader.conf`.

```
services:
  windows:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - c:\cfn\cfn-hup.conf
        - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Vous pouvez gérer n'importe quel autre service Windows de la même manière en utilisant son nom interne (et non son nom d'affichage) comme référence.

Résolution des problèmes de création de pile

Si votre pile échoue lors de la création, le comportement par défaut consiste à effectuer une restauration en cas d'échec. Bien que ce comportement par défaut soit généralement approprié, car

il permet d'éviter des frais inutiles, il complique le débogage du problème qui est à l'origine de l'échec de création de la pile.

Pour désactiver ce comportement lors de la création ou de la mise à jour de votre stack avec la CloudFormation console, choisissez l'option Conserver les ressources correctement provisionnées sous Options d'échec de la pile. Pour de plus amples informations, veuillez consulter [Choisissez comment gérer les défaillances lors de l'approvisionnement des ressources](#). Cette option vous permet d'ouvrir une session sur l'instance et d'examiner les fichiers journaux afin d'identifier les problèmes rencontrés lors de l'exécution des scripts de démarrage.

Voici les principaux journaux que vous devez examiner :

- Le journal de configuration EC2 à l'adresse %ProgramData%\Amazon\EC2Launch\log\agent.log
- Le journal cfn-init dans C:\cfn\log\cfn-init.log (vérifiez les codes de sortie et messages d'erreur pour localiser précisément la défaillance)

Pour plus de journaux, consultez les rubriques suivantes du Guide de l'utilisateur Amazon EC2 :

- [EC2Launchstructure du répertoire](#)
- [Structure du répertoire EC2Launch v2](#)

Pour plus d'informations sur la résolution des problèmes d'amorçage, consultez [Comment résoudre les problèmes liés aux scripts d'assistance qui ne démarrent pas dans une pile contenant des instances](#) ? CloudFormation Windows .

Étendez les fonctionnalités de votre modèle avec les CloudFormation types de ressources fournis

CloudFormation propose plusieurs types de ressources que vous pouvez utiliser dans votre modèle de pile pour étendre ses fonctionnalités au-delà de celles d'un simple modèle de pile.

Ces types de ressources incluent :

Type de ressource	Description	Documentation
Ressources personnalisées	Le type de <code>AWS::CloudFormation::CustomResource</code> ressource vous permet de créer des ressources personnalisées qui peuvent effectuer des tâches de provisionnement spécifiques ou inclure des ressources qui ne sont pas disponibles en tant que types de CloudFormation ressources.	Ressources personnalisées
Macros	Le type de <code>AWS::CloudFormation::Macro</code> ressource définit un code réutilisable qui peut effectuer un traitement personnalisé sur des CloudFormation modèles. Les macros peuvent modifier vos modèles, générer des ressources supplémentaires ou effectuer d'autres opérations personnalisées lors de la création ou de la mise à jour de la pile.	Macros de modèle
Piles imbriquées	Le type de <code>AWS::CloudFormation::Stack</code> ressource vous permet de créer des piles imbriquées dans vos CloudFormation modèles pour des architectures de pile plus modulaires et réutilisables.	Piles imbriquées
StackSet	Le type de <code>AWS::CloudFormation::StackSet</code> ressource crée ou met à jour un CloudFormation StackSet, qui est un conteneur pour les piles pouvant être déployées dans plusieurs Comptes AWS régions.	Gérer les piles avec StackSets
Condition d'attente	Le type de ressource <code>AWS::CloudFormation::WaitCondition</code> met en pause la création ou la mise à jour d'une pile jusqu'à ce qu'une condition spécifique soit remplie,	Conditions d'attente

Type de ressource	Description	Documentation
	comme la réussite d'un processus long ou la disponibilité de ressources externes.	
Gestionnaire de condition d'attente	Le type de ressource <code>AWS::CloudFormation::WaitConditionHandle</code> fonctionne avec le type de ressource <code>AWS::CloudFormation::WaitCondition</code> . Il fournit une URL présignée qui est utilisée pour envoyer des signaux indiquant qu'une condition spécifique a été remplie. Ces signaux permettent au processus de création ou de mise à jour de la pile de se poursuivre.	Conditions d'attente

Créer une logique de provisionnement personnalisée avec des ressources personnalisées

Les ressources personnalisées vous permettent d'écrire une logique de provisionnement personnalisée dans vos CloudFormation modèles et de l' CloudFormation exécuter chaque fois que vous créez, mettez à jour (si vous avez modifié la ressource personnalisée) ou supprimez une pile. Cela peut être utile lorsque vos exigences de provisionnement impliquent une logique complexe ou des flux de travail qui ne peuvent pas être exprimés avec les types CloudFormation de ressources intégrés.

Par exemple, vous souhaitez peut-être inclure des ressources qui ne sont pas disponibles en tant que types de CloudFormation ressources. Vous pouvez inclure ces ressources à l'aide de ressources personnalisées. De cette façon, vous pouvez continuer à gérer toutes les ressources connexes dans une seule pile.

Pour définir une ressource personnalisée dans votre CloudFormation modèle, vous utilisez le type de Custom : `MyCustomResourceTypeName` ressource `AWS::CloudFormation::CustomResource` ou. Les ressources personnalisées nécessitent une propriété, le jeton de service, qui indique à qui les demandes sont CloudFormation envoyées, par exemple une rubrique Amazon SNS ou une fonction Lambda.

Les rubriques suivantes fournissent des informations sur l'utilisation des ressources personnalisées.

Rubriques

- [Fonctionnement des ressources personnalisées](#)
- [Délai d'attente de réponse](#)
- [CloudFormation demande de ressource personnalisée et référence de réponse](#)
- [Ressources personnalisées prises en charge par Amazon SNS](#)
- [Ressources personnalisées basées sur Lambda](#)

Note

Le CloudFormation registre et les ressources personnalisées offrent chacun leurs propres avantages. Les ressources personnalisées offrent les avantages suivants :

- Vous n'avez pas besoin d'enregistrer la ressource.
- Vous pouvez inclure une ressource entière dans un modèle sans l'enregistrer.
- Prise en charge des opérations Create, Update et Delete

Les ressources basées sur le registre offrent les avantages suivants :

- La prise en charge de la modélisation, du provisionnement et de la gestion des ressources d'applications tierces
- La prise en charge des opérations Create, Read, Update, Delete et List (CRUDL)
- La prise en charge de la détection de l'écart sur les types de ressources privées et tierces

Contrairement aux ressources personnalisées, les ressources basées sur le registre n'ont pas besoin d'associer une rubrique Amazon SNS ou une fonction Lambda pour effectuer des opérations CRUDL. Pour de plus amples informations, veuillez consulter [Gestion des extensions avec le CloudFormation registre](#).

Fonctionnement des ressources personnalisées

Le processus général de configuration d'une nouvelle ressource personnalisée comprend les étapes suivantes. Ces étapes impliquent deux rôles : le fournisseur de ressources personnalisées qui possède la ressource personnalisée et le développeur de modèles qui crée un modèle incluant un

type de ressource personnalisée. Il peut s'agir de la même personne, mais si ce n'est pas le cas, le fournisseur de ressources personnalisées doit collaborer avec le développeur de modèles.

1. Le fournisseur de ressources personnalisées écrit une logique qui détermine la manière de traiter les demandes émanant de la ressource personnalisée CloudFormation et d'exécuter des actions sur celle-ci.
2. Le fournisseur de ressources personnalisées crée la rubrique Amazon SNS ou la fonction Lambda à laquelle il CloudFormation peut envoyer des demandes. La rubrique Amazon SNS ou la fonction Lambda doit se trouver dans la même région que celle où la pile sera créée.
3. Le fournisseur de ressources personnalisées communique l'ARN de la rubrique Amazon SNS ou l'ARN de la fonction Lambda au développeur de modèles.
4. Le développeur du modèle définit la ressource personnalisée dans son CloudFormation modèle. Cela inclut un jeton de service et tous les paramètres de données d'entrée. Le jeton de service et la structure des données d'entrée sont définis par le fournisseur de ressources personnalisées. Le jeton de service spécifie l'ARN de la rubrique Amazon SNS ou l'ARN de la fonction Lambda et est toujours requis, mais les données d'entrée sont facultatives selon la ressource personnalisée.

Désormais, chaque fois que quelqu'un utilise le modèle pour créer, mettre à jour ou supprimer la ressource personnalisée, CloudFormation envoie une demande au jeton de service spécifié, puis attend une réponse avant de poursuivre l'opération de pile.

Voici un résumé du processus de création d'une pile à partir du modèle :

1. CloudFormation envoie une demande au jeton de service spécifié. La demande inclut des informations telles que le type de demande et une URL de compartiment Amazon S3 pré-signée, à laquelle la ressource personnalisée envoie des réponses. Pour plus d'informations sur les éléments inclus dans la requête, consultez [CloudFormation demande de ressource personnalisée et référence de réponse](#).

Les exemples de données suivants montrent ce qui est CloudFormation inclus dans une Create demande. Dans cet exemple, ResourceProperties permet CloudFormation de créer une charge utile personnalisée à envoyer à la fonction Lambda.

```
{
  "RequestType" : "Create",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
```

```
"ResponseURL" : "http://pre-signed-S3-url-for-response",
"ResourceType" : "Custom::TestResource",
"LogicalResourceId" : "MyTestResource",
"ResourceProperties" : {
  "Name" : "Value",
  "List" : [ "1", "2", "3" ]
}
```

2. Le fournisseur de ressources personnalisées traite la CloudFormation demande et renvoie une réponse FAILED à l'URL pré-signée. SUCCESS Le custom resource provider fournit la réponse dans un fichier au format JSON et la télécharge vers l'URL S3 pré-signée. Pour plus d'informations, consultez la section [Chargement d'objets présignés URLs](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Dans cette réponse, le custom resource provider peut aussi inclure les paires nom-valeur auxquelles le template developer peut accéder. Par exemple, la réponse peut inclure les données de sortie si la requête a abouti ou un message d'erreur en cas d'échec. Pour plus d'informations sur les réponses, consultez [CloudFormation demande de ressource personnalisée et référence de réponse](#).

Important

Si les paires nom-valeur contiennent des informations sensibles, vous devez utiliser le champ NoEcho pour masquer la sortie de la ressource personnalisée. Dans le cas contraire, les valeurs sont visibles à travers APIs les valeurs des propriétés de cette surface (telles que `DescribeStackEvents`).

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Le custom resource provider est chargé d'écouter la requête et d'y répondre. Par exemple, pour les notifications Amazon SNS, le fournisseur de ressources personnalisées doit écouter et répondre aux notifications envoyées à un ARN de rubrique spécifique. CloudFormation attend et écoute une réponse à l'emplacement de l'URL pré-signée.

L'échantillon de données suivant montre ce qu'une ressource personnalisée peut inclure dans une réponse :

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MyTestResource",
  "PhysicalResourceId" : "TestResource1",
  "Data" : {
    "OutputName1" : "Value1",
    "OutputName2" : "Value2",
  }
}
```

3. Après avoir obtenu une SUCCESS réponse, CloudFormation poursuit l'opération de pile. Sans réponse ou si une réponse FAILED est renvoyée, l'opération échoue. Les données de sortie de la ressource personnalisée sont stockées à l'emplacement de l'URL pré-signée. Le développeur du modèle peut récupérer ces données à l'aide de la GetAtt fonction [Fn::](#).

Note

Si vous en utilisez AWS PrivateLink, les ressources personnalisées du VPC doivent avoir accès à des compartiments S3 CloudFormation spécifiques. Les ressources personnalisées doivent envoyer des réponses à une URL Amazon S3 pré-signée. S'ils ne peuvent pas envoyer de réponses à Amazon S3, ils CloudFormation ne recevront pas de réponse et l'opération de stack échouera. Pour de plus amples informations, veuillez consulter [Accès CloudFormation via un point de terminaison d'interface \(AWS PrivateLink\)](#).

Délai d'attente de réponse

Le délai d'attente par défaut pour votre ressource personnalisée est de 3600 secondes (1 heure). Si aucune réponse n'est reçue pendant ce délai, l'opération de pile échoue.

Vous pouvez ajuster la valeur du délai d'attente en fonction du temps que vous pensez que la réponse de la ressource personnalisée prendra. Par exemple, lors du provisionnement d'une ressource personnalisée qui invoque une fonction Lambda qui devrait répondre dans les cinq minutes, vous pouvez définir un délai d'attente de cinq minutes dans le modèle de pile en spécifiant la propriété `ServiceTimeout`. Pour de plus amples informations, veuillez consulter [CloudFormation](#)

[demande de ressource personnalisée et référence de réponse](#). Ainsi, si une erreur dans la fonction Lambda provoque son blocage, l'opération de pile CloudFormation échouera au bout de cinq minutes au lieu d'attendre une heure complète.

Cependant, veuillez à ne pas définir une valeur de délai d'expiration trop faible. Pour éviter les délais d'expiration inattendus, assurez-vous que votre ressource personnalisée dispose de suffisamment de temps pour effectuer les actions nécessaires et renvoyer une réponse.

CloudFormation demande de ressource personnalisée et référence de réponse

CloudFormation gère les ressources personnalisées via un protocole de demande-réponse qui communique avec votre fournisseur de ressources personnalisées. Chaque demande inclut un type de demande (`CreateUpdate`, ou `Delete`) et suit ce flux de travail de haut niveau :

1. Un développeur de modèles définit une ressource personnalisée avec un `ServiceToken` et `ServiceTimeout` dans le modèle et lance une opération de pile.
2. CloudFormation envoie une demande JSON au fournisseur de ressources personnalisées via SNS ou Lambda.
3. Le fournisseur de ressources personnalisées traite la demande et renvoie une réponse JSON à une URL de compartiment Amazon S3 présignée avant l'expiration du délai d'expiration.
4. CloudFormation lit la réponse et poursuit l'opération de pile. Si aucune réponse n'est reçue avant la fin du délai imparti, la demande est considérée comme infructueuse et l'opération de pile échoue.

Pour de plus amples informations, veuillez consulter [Fonctionnement des ressources personnalisées](#).

Cette section décrit la structure, les paramètres et les réponses attendues pour chaque type de demande.

Note

La taille totale du corps de la réponse ne peut pas dépasser 4 096 octets.

Configuration du modèle

Lors de la définition d'une ressource personnalisée dans un modèle, le développeur du modèle utilise [AWS::CloudFormation::CustomResource](#) les propriétés suivantes :

ServiceToken

Un ARN de rubrique Amazon SNS ou un ARN de fonction Lambda provenant de la même région que la pile.

Obligatoire : oui

Type : chaîne

ServiceTimeout

Durée maximale, en secondes, avant l'expiration d'une opération sur une ressource personnalisée. Il doit s'agir d'une valeur comprise entre 1 et 3 600. Par défaut : 3 600 secondes (1 heure).

Obligatoire : non

Type : chaîne

Des propriétés de ressources supplémentaires sont prises en charge. Les propriétés des ressources seront incluses comme `ResourceProperties` dans la demande. Le fournisseur de ressources personnalisées doit déterminer quelles propriétés sont valides et quelles sont leurs valeurs acceptables.

Objet Requête

Create

Lorsque le développeur du modèle crée une pile contenant une ressource personnalisée, il CloudFormation envoie une demande avec `RequestType` set to `Create`.

Les requêtes de création contiennent les champs suivants :

RequestType

`Create`.

Obligatoire : oui

Type : chaîne

RequestId

Un ID unique pour la demande.

La combinaison de `StackId` avec `RequestId` forme une valeur que vous pouvez utiliser pour identifier de manière unique une demande sur une ressource personnalisée spécifique.

Obligatoire : oui

Type : chaîne

`StackId`

L'Amazon Resource Name (ARN) qui identifie la pile contenant la ressource personnalisée.

La combinaison de `StackId` avec `RequestId` forme une valeur que vous pouvez utiliser pour identifier de manière unique une demande sur une ressource personnalisée spécifique.

Obligatoire : oui

Type : chaîne

`ResponseURL`

L'URL de réponse identifie un compartiment S3 présigné qui reçoit les réponses du fournisseur de ressources personnalisées à CloudFormation.

Obligatoire : oui

Type : chaîne

`ResourceType`

Type de ressource personnalisée choisi par le développeur de modèle dans le modèle CloudFormation . Le nom des types de ressources personnalisées ne doit pas dépasser 60 caractères. Il peut inclure des caractères alphanumériques, ainsi que les caractères suivants : `_@-`.

Obligatoire : oui

Type : chaîne

`LogicalResourceId`

Le nom, choisi par le développeur du modèle, (ID logique) de la ressource personnalisée dans le modèle CloudFormation .

Obligatoire : oui

Type : chaîne

ResourceProperties

Ce champ contient le contenu de l'objet `Properties` envoyé par le développeur du modèle. Son contenu est défini par le custom resource provider.

Obligatoire : non

Type : objet JSON

Exemple

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-create-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Update

Lorsque le développeur du modèle modifie les propriétés d'une ressource personnalisée dans le modèle et met à jour la pile, CloudFormation envoie une demande au fournisseur de ressources personnalisées avec la valeur `RequestType` définie sur `Update`. Cela signifie que le code de votre ressource personnalisée n'a pas besoin de détecter les modifications dans les propriétés, puisqu'il sait que celles-ci ont été modifiées lorsque le type de requête est `Update`.

Les demandes de mise à jour contiennent les champs suivants :

RequestType

`Update`.

Obligatoire : oui

Type : chaîne

RequestId

Un ID unique pour la demande.

La combinaison de `StackId` avec `RequestId` forme une valeur que vous pouvez utiliser pour identifier de manière unique une demande sur une ressource personnalisée spécifique.

Obligatoire : oui

Type : chaîne

StackId

L'Amazon Resource Name (ARN) qui identifie la pile contenant la ressource personnalisée.

La combinaison de `StackId` avec `RequestId` forme une valeur que vous pouvez utiliser pour identifier de manière unique une demande sur une ressource personnalisée spécifique.

Obligatoire : oui

Type : chaîne

ResponseURL

L'URL de réponse identifie un compartiment S3 présigné qui reçoit les réponses du fournisseur de ressources personnalisées à CloudFormation.

Obligatoire : oui

Type : chaîne

ResourceType

Type de ressource personnalisée choisi par le développeur de modèle dans le modèle CloudFormation . Le nom des types de ressources personnalisées ne doit pas dépasser 60 caractères. Il peut inclure des caractères alphanumériques, ainsi que les caractères suivants : `_@-`. Vous ne pouvez pas modifier le type pendant une mise à jour.

Obligatoire : oui

Type : chaîne

LogicalResourceId

Le nom, choisi par le développeur du modèle, (ID logique) de la ressource personnalisée dans le modèle CloudFormation .

Obligatoire : oui

Type : chaîne

PhysicalResourceId

Un identifiant physique personnalisé défini par le fournisseur de ressources qui est unique pour ce fournisseur.

Obligatoire : oui

Type : chaîne

ResourceProperties

Ce champ contient le contenu de l'objet `Properties` envoyé par le développeur du modèle. Son contenu est défini par le custom resource provider.

Obligatoire : non

Type : objet JSON

OldResourceProperties

Utilisé uniquement pour les demandes `Update`. Valeurs des propriétés de ressource qui ont été déclarées précédemment par le template `developper` dans le modèle `CloudFormation`.

Obligatoire : oui

Type : objet JSON

Exemple

```
{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-update-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "new-string",
    "key2" : [ "new-list" ],
```

```
    "key3" : { "key4" : "new-map" }
  },
  "OldResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Delete

Lorsque le développeur du modèle supprime la pile ou supprime la ressource personnalisée du modèle, puis met à jour la pile, CloudFormation envoie une demande avec `RequestType` set `ToDelete`.

Les requêtes de suppression contiennent les champs suivants :

RequestType

Delete.

Obligatoire : oui

Type : chaîne

RequestId

Un ID unique pour la demande.

Obligatoire : oui

Type : chaîne

StackId

L'Amazon Resource Name (ARN) qui identifie la pile contenant la ressource personnalisée.

Obligatoire : oui

Type : chaîne

ResponseURL

L'URL de réponse identifie un compartiment S3 présigné qui reçoit les réponses du fournisseur de ressources personnalisées à CloudFormation.

Obligatoire : oui

Type : chaîne

ResourceType

Type de ressource personnalisée choisi par le développeur de modèle dans le modèle CloudFormation . Le nom des types de ressources personnalisées ne doit pas dépasser 60 caractères. Il peut inclure des caractères alphanumériques, ainsi que les caractères suivants : _@- .

Obligatoire : oui

Type : chaîne

LogicalResourceId

Le nom, choisi par le développeur du modèle, (ID logique) de la ressource personnalisée dans le modèle CloudFormation .

Obligatoire : oui

Type : chaîne

PhysicalResourceId

Un identifiant physique personnalisé défini par le fournisseur de ressources qui est unique pour ce fournisseur.

Obligatoire : oui

Type : chaîne

ResourceProperties

Ce champ contient le contenu de l'objet Properties envoyé par le développeur du modèle. Son contenu est défini par le custom resource provider.

Obligatoire : non

Type : objet JSON

Exemple

```
{
```

```
"RequestType" : "Delete",
"RequestId" : "unique-request-id",
"StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
"ResponseURL" : "pre-signed-url-for-delete-response",
"ResourceType" : "Custom::MyCustomResourceType",
"LogicalResourceId" : "resource-logical-id",
"PhysicalResourceId" : "provider-defined-physical-id",
"ResourceProperties" : {
  "key1" : "string",
  "key2" : [ "list" ],
  "key3" : { "key4" : "map" }
}
}
```

Objet Réponse

Le fournisseur de ressources personnalisées envoie une réponse à l'URL pré-signée pour tous les types de demandes. Si le fournisseur de ressources personnalisées n'envoie pas de réponse, CloudFormation attend la fin de l'opération.

La réponse doit être un objet JSON avec les champs suivants :

Status

Doit être SUCCESS ou FAILED.

Obligatoire : oui

Type : chaîne

RequestId

Un ID unique pour la demande. Copiez cette valeur exactement telle qu'elle apparaît dans la demande.

Obligatoire : oui

Type : chaîne

StackId

L'Amazon Resource Name (ARN) qui identifie la pile contenant la ressource personnalisée. Copiez cette valeur exactement telle qu'elle apparaît dans la demande.

Obligatoire : oui

Type : chaîne

LogicalResourceId

Le nom, choisi par le développeur du modèle, (ID logique) de la ressource personnalisée dans le modèle CloudFormation . Copiez cette valeur exactement telle qu'elle apparaît dans la demande.

Obligatoire : oui

Type : chaîne

PhysicalResourceId

Cette valeur doit être un identifiant unique pour le fournisseur de ressources personnalisées et sa taille ne peut pas dépasser 1 Ko. La valeur doit être une chaîne non vide et doit être identique pour toutes les réponses pour la même ressource.

Lors de la mise à jour de ressources personnalisées, la valeur renvoyée pour `PhysicalResourceId` détermine le comportement de mise à jour. Si la valeur reste la même, CloudFormation considère qu'il s'agit d'une mise à jour normale. Si la valeur change, CloudFormation interprète la mise à jour comme un remplacement et envoie une demande de suppression à l'ancienne ressource. Pour de plus amples informations, veuillez consulter [AWS::CloudFormation::CustomResource](#).

Obligatoire : oui

Type : chaîne

Reason

Décrit le motif d'une réponse à un échec.

Obligatoire si `Status` a pour valeur `FAILED`. Sinon, c'est facultatif.

Obligatoire : Conditionnelle

Type : chaîne

NoEcho

Indique s'il faut masquer la sortie de la ressource personnalisée lorsque celle-ci est récupérée à l'aide de la fonction `Fn::GetAtt`. Si la valeur est définie sur `true`, toutes les valeurs renvoyées sont masquées par des astérisques (`*****`), à l'exception de celles stockées dans la section

Metadata du modèle. CloudFormation ne transforme pas, ne modifie pas et ne censure aucune information que vous incluez dans la section Metadata. La valeur par défaut est `false`.

Pour plus d'informations sur l'utilisation de `NoEcho` pour masquer les informations sensibles, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Disponible uniquement pour `Create` et pour `Update` les réponses. Non pris en charge pour `Delete` les réponses.

Obligatoire : non

Type : valeur booléenne

Data

Les paires nom-valeur définies par le custom resource provider à envoyer avec la réponse. Vous pouvez accéder aux valeurs fournies ici par nom dans le modèle avec `Fn::GetAtt`.

Disponible uniquement pour `Create` et pour `Update` les réponses. Non pris en charge pour `Delete` les réponses.

Important

Si les paires nom-valeur contiennent des informations sensibles, vous devez utiliser le champ `NoEcho` pour masquer la sortie de la ressource personnalisée. Dans le cas contraire, les valeurs sont visibles à travers APIs les valeurs des propriétés de cette surface (telles que `DescribeStackEvents`).

Obligatoire : non

Type : objet JSON

Exemples de réponses réussies

Create et Update réponse

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
```



```
"PhysicalResourceId": "provider-defined-physical-id",
"NoEcho": true,
"Data": {
  "key1": "value1",
  "key2": "value2"
}
```

Réponse d>Delete

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id"
}
```

Exemple de réponse échouée

```
{
  "Status": "FAILED",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "Reason": "Required failure reason string"
}
```

Ressources personnalisées prises en charge par Amazon SNS

La rubrique suivante explique comment configurer une ressource personnalisée avec un jeton de service qui indique la rubrique Amazon SNS à laquelle les demandes sont CloudFormation envoyées. Vous découvrirez également la séquence d'événements et de messages envoyés et reçus à la suite de la création, de la mise à jour et de la suppression d'une pile de ressources personnalisées.

Les ressources personnalisées et Amazon SNS vous permettent d'activer des scénarios, tels que l'ajout de nouvelles ressources à une pile et l'injection de données dynamiques dans une pile. Par exemple, lorsque vous créez une pile, vous CloudFormation pouvez envoyer une Create demande à un sujet surveillé par une application exécutée sur une instance Amazon EC2. La notification Amazon SNS incite l'application à exécuter les tâches d'allocation supplémentaires, telles que la récupération

d'un groupe d'adresses IP Elastic autorisées. Une fois cela fait, l'application envoie une réponse (et toutes les données de sortie) qui indique CloudFormation de poursuivre l'opération de pile.

Lorsque vous spécifiez une rubrique Amazon SNS comme cible d'une ressource personnalisée, CloudFormation vous envoie des messages à la rubrique SNS spécifiée pendant les opérations de stack impliquant la ressource personnalisée. Pour traiter ces messages et effectuer les actions nécessaires, vous devez disposer d'un point de terminaison pris en charge abonné à la rubrique SNS.

Pour une présentation des ressources personnalisées et de leur fonctionnement, consultez [Fonctionnement des ressources personnalisées](#). Pour plus d'informations sur Amazon SNS et son fonctionnement, consultez le [Guide du développeur Amazon Simple Notification Service](#).

Utilisation d'Amazon SNS pour créer des ressources personnalisées

Rubriques

- [Étape 1 : Création de la pile](#)
- [Étape 2 : Mises à jour d'une pile](#)
- [Étape 3 : Suppression d'une pile](#)

Étape 1 : Création de la pile

1. Le développeur du modèle crée une CloudFormation pile contenant une ressource personnalisée.

Dans l'exemple de modèle ci-dessous, nous utilisons le nom de type de ressource personnalisée `Custom::SeleniumTester` pour la ressource personnalisée avec l'ID logique `MySeleniumTest`. Le nom des types de ressources personnalisées doit être alphanumérique et ne doit pas dépasser 60 caractères.

Le type de ressource personnalisé est déclaré avec un jeton de service, des propriétés facultatives spécifiques au fournisseur et des GetAtt attributs [Fn::](#) facultatifs définis par le fournisseur de ressources personnalisé. Ces propriétés et ces attributs peuvent être utilisés pour transmettre les informations du template developper au custom resource provider, et vice versa. Le jeton de service spécifie une rubrique Amazon SNS que le fournisseur de ressources a configuré.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
```

```

    "Type": "Custom::SeleniumTester",
    "Version" : "1.0",
    "Properties" : {
        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
            "http://search.mysite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
    }
},
"Outputs" : {
    "topItem" : {
        "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    },
    "numRespondents" : {
        "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
    }
}
}

```

Note

Les noms et les valeurs des données accessibles avec `Fn::GetAtt` sont renvoyés par le fournisseur de ressources personnalisées lors de la réponse du fournisseur à CloudFormation. Si le custom resource provider est un tiers, le template developer doit obtenir les noms de ces valeurs de retour auprès du custom resource provider.

2. CloudFormation envoie une notification Amazon SNS au fournisseur de ressources "RequestType" : "Create" contenant des informations sur la pile, les propriétés de ressources personnalisées issues du modèle de pile et une URL S3 pour la réponse.

La rubrique SNS qui est utilisée pour envoyer la notification est intégrée dans la propriété `ServiceToken` du modèle. Pour éviter les valeurs codées en dur, un template developer peut utiliser un paramètre de modèle de sorte que la valeur soit entrée lors du lancement de la pile.

L'exemple suivant présente une requête de création de ressource personnalisée (Create), qui inclut un nom de type de ressource personnalisée, `Custom::SeleniumTester`, créé avec la valeur `LogicalResourceId MySeleniumTester` :

```
{
```

```

    "RequestType" : "Create",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
    "ResponseURL" : "http://pre-signed-S3-url-for-response",
    "ResourceType" : "Custom::SeleniumTester",
    "LogicalResourceId" : "MySeleniumTester",
    "ResourceProperties" : {
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
    }
}

```

Pour plus d'informations sur l'objet de demande pour les demandes Create, consultez la rubrique [Référence de demande et de réponse](#).

3. Le custom resource provider traite les données envoyées par le template developper et détermine si la demande Create a réussi. Le fournisseur de ressources utilise ensuite l'URL S3 envoyée par CloudFormation pour envoyer une réponse de l'un SUCCESS ou de l'autre FAILED.

En fonction du type de réponse, différents champs de réponse sont attendues par CloudFormation. Pour plus d'informations sur les champs de réponse pour un type de demande particulier, veuillez consulter la documentation relative à ce type de demande dans la section [Référence de demande et de réponse](#).

En réponse à une demande de création ou de mise à jour, le fournisseur de ressources personnalisées peut renvoyer des éléments de données dans le champ Data de la réponse. Il s'agit de paires nom-valeur, où les noms correspondent aux attributs Fn::GetAtt utilisés avec la ressource personnalisée dans le modèle de la pile. Les valeurs sont les données qui sont renvoyées quand le template developper appelle Fn::GetAtt au niveau de la ressource avec le nom d'attribut.

Voici un exemple de réponse de ressource personnalisée :

```

{
    "Status" : "SUCCESS",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",

```

```
"LogicalResourceId" : "MySeleniumTester",
"PhysicalResourceId" : "Tester1",
"Data" : {
  "resultsPage" : "http://www.myexampledomain/test-results/guid",
  "lastUpdate" : "2012-11-14T03:30Z"
}
```

Pour plus d'informations sur l'objet de réponse pour les demandes `Create`, consultez la rubrique [Référence de demande et de réponse](#).

Les champs `StackId`, `RequestId` et `LogicalResourceId` doivent être copiés mot pour mot à partir de la demande.

4. CloudFormation déclare le statut de la pile comme `CREATE_COMPLETE` ou `CREATE_FAILED`. Si la pile a été créée avec succès, le développeur du modèle peut utiliser les valeurs de sortie de la ressource personnalisée créée en y accédant avec [Fn::GetAtt](#).

Par exemple, le modèle de ressource personnalisée utilisé à titre d'illustration a eu recours à `Fn::GetAtt` pour copier les sorties des ressources dans les sorties de la pile :

```
"Outputs" : {
  "topItem" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
  },
  "numRespondents" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
  }
}
```

Étape 2 : Mises à jour d'une pile

Pour mettre à jour une pile existante, vous devez présenter un modèle qui spécifie les mises à jour des propriétés de ressources de la pile, comme illustré dans l'exemple ci-dessous. CloudFormation met à jour uniquement les ressources dont des modifications sont spécifiées dans le modèle. Pour de plus amples informations, veuillez consulter [Comprendre les comportements de mise à jour des ressources d'une pile](#).

Vous pouvez mettre à jour les ressources personnalisées qui nécessitent un remplacement de la ressource physique sous-jacente. Lorsque vous mettez à jour une ressource personnalisée dans

un CloudFormation modèle, CloudFormation envoie une demande de mise à jour à cette ressource personnalisée. Si une ressource personnalisée nécessite un remplacement, la nouvelle ressource personnalisée doit envoyer une réponse avec le nouvel ID physique. Quand CloudFormation reçoit la réponse, il compare la valeur `PhysicalResourceId` entre l'ancienne ressource personnalisée et la nouvelle. S'ils sont différents, CloudFormation reconnaît la mise à jour comme un remplacement et envoie une demande de suppression à l'ancienne ressource, comme indiqué dans [Étape 3 : Suppression d'une pile](#).

Note

Si vous n'avez pas apporté de modifications à la ressource personnalisée, vous CloudFormation ne lui enverrez pas de demandes lors d'une mise à jour de la pile.

1. Le template developper lance une mise à jour de la pile qui contient une ressource personnalisée. Pendant une mise à jour, le template developper peut spécifier de nouvelles propriétés dans le modèle de la pile.

Voici un exemple de mise à jour (Update) apportée au modèle de la pile à l'aide d'un type de ressource personnalisée :

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
      "Type": "Custom::SeleniumTester",
      "Version" : "1.0",
      "Properties" : {
        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
          "http://search.mysite.com",
          "http://mynewsite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
      }
    }
  },
  "Outputs" : {
    "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    }
  },
}
```

```

    "numRespondents" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
    }
  }
}

```

- CloudFormation envoie une notification Amazon SNS au fournisseur de ressources contenant des informations similaires à celles de l'CreateAppel, sauf que le OldResourceProperties champ contient les anciennes propriétés des ressources et ResourceProperties contient les propriétés des ressources mises à jour (le cas échéant). "RequestType" : "Update"

Voici un exemple de requête Update :

```

{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com",
      "http://mynewsite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
  },
  "OldResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}

```

Pour plus d'informations sur l'objet de demande pour les demandes Update, consultez la rubrique [Référence de demande et de réponse](#).

- Le fournisseur de ressources personnalisées traite les données envoyées par CloudFormation. La ressource personnalisée effectue la mise à jour et envoie une réponse à l'URL S3 SUCCESS

ou FAILED à cette adresse. CloudFormation compare ensuite les `PhysicalResourceIDs` anciennes et les nouvelles ressources personnalisées. S'ils sont différents, CloudFormation reconnaît que la mise à jour doit être remplacée et envoie une demande de suppression à l'ancienne ressource. L'exemple suivant présente la réponse du custom resource provider à une demande Update.

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester2"
}
```

Pour plus d'informations sur l'objet de réponse pour les demandes Update, consultez la rubrique [Référence de demande et de réponse](#).

Les champs `StackId`, `RequestId` et `LogicalResourceId` doivent être copiés mot pour mot à partir de la demande.

4. CloudFormation déclare le statut de la pile comme `UPDATE_COMPLETE` ou `UPDATE_FAILED`. En cas d'échec de la mise à jour, la pile est restaurée. Si la pile a bien été mise à jour, le template developer peut accéder aux nouvelles valeurs de sortie de la ressource personnalisée créée avec `Fn::GetAtt`.

Étape 3 : Suppression d'une pile

1. Le template developer supprime une pile qui contient une ressource personnalisée. CloudFormation récupère les propriétés spécifiées dans le modèle de la pile, ainsi que la rubrique SNS, et se prépare à envoyer une demande au custom resource provider.
2. CloudFormation envoie une notification Amazon SNS au fournisseur de ressources "RequestType" : "Delete" contenant des informations actuelles sur la pile, les propriétés de ressources personnalisées issues du modèle de pile et une URL S3 pour la réponse.

Chaque fois que vous supprimez une pile ou effectuez une mise à jour qui supprime ou remplace la ressource personnalisée CloudFormation, `PhysicalResourceId` comparez les anciennes et les nouvelles ressources personnalisées. S'ils sont différents, CloudFormation reconnaît la mise à

jour comme un remplacement et envoie une demande de suppression pour l'ancienne ressource (OldPhysicalResource), comme illustré dans l'exemple de Delete demande suivant.

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
    "http://mynewsite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
  }
}
```

Pour plus d'informations sur l'objet de demande pour les demandes Delete, consultez la rubrique [Référence de demande et de réponse](#).

DescribeStackResource, DescribeStackResources, et ListStackResources affiche le nom défini par l'utilisateur, s'il a été spécifié.

3. Le fournisseur de ressources personnalisées traite les données envoyées par CloudFormation et détermine si la Delete demande a abouti. Le fournisseur de ressources utilise ensuite l'URL S3 envoyée par CloudFormation pour envoyer une réponse de l'un SUCCESS ou de l'autre FAILED. Pour supprimer une pile avec une ressource personnalisée, le custom resource provider doit répondre avec succès à une demande de suppression.

Voici un exemple de réponse du custom resource provider à une demande Delete :

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1"
```

```
}
```

Pour plus d'informations sur l'objet de réponse pour les demandes Delete, consultez la rubrique [Référence de demande et de réponse](#).

Les champs `StackId`, `RequestId` et `LogicalResourceId` doivent être copiés mot pour mot à partir de la demande.

4. CloudFormation déclare le statut de la pile comme `DELETE_COMPLETE` ou `DELETE_FAILED`.

Ressources personnalisées basées sur Lambda

Lorsque vous associez une fonction Lambda à une ressource personnalisée, la fonction est appelée chaque fois que la ressource personnalisée est créée, mise à jour ou supprimée. CloudFormation appelle une API Lambda pour appeler la fonction et pour transmettre toutes les données de demande (comme le type de demande et les propriétés des ressources) à la fonction. CloudFormation Combinées à la puissance et à la personnalisation des fonctions Lambda, elles permettent un large éventail de scénarios, tels que la recherche dynamique d' IDs AMI lors de la création d'une pile ou la mise en œuvre et l'utilisation de fonctions utilitaires, telles que les fonctions d'inversion de chaînes de caractères.

Pour une présentation des ressources personnalisées et de leur fonctionnement, consultez [Fonctionnement des ressources personnalisées](#).

Rubriques

- [Démonstration : créer un mécanisme de délai avec une ressource personnalisée prise en charge par Lambda](#)
- [Module cfn-response](#)

Démonstration : créer un mécanisme de délai avec une ressource personnalisée prise en charge par Lambda

Cette procédure pas à pas vous explique comment configurer et lancer une ressource personnalisée basée sur Lambda à l'aide d'un exemple de modèle. CloudFormation Ce modèle crée un mécanisme de délai qui suspend les déploiements de pile pendant une durée spécifiée. Cela peut être utile lorsque vous devez introduire des délais délibérés pendant l'approvisionnement en ressources, par exemple lorsque vous attendez que les ressources se stabilisent avant de créer des ressources dépendantes.

 Note

Alors que les ressources personnalisées soutenues par Lambda étaient auparavant recommandées pour récupérer les AMI IDs, nous recommandons désormais d'utiliser des paramètres. AWS Systems Manager Cette approche rend vos modèles plus réutilisables et plus faciles à maintenir. Pour de plus amples informations, veuillez consulter [Obtenir une valeur en texte brut depuis le magasin de paramètres Systems Manager](#).

Rubriques

- [Présentation de](#)
- [Exemple de modèle](#)
- [Démonstration du modèle d'exemple](#)
- [Conditions préalables](#)
- [Lancement de la pile](#)
- [Nettoyage des ressources](#)
- [Informations connexes](#)

Présentation de

Le modèle de pile utilisé dans cette démonstration crée une ressource personnalisée basée sur Lambda. Cette ressource personnalisée introduit un délai configurable (60 secondes par défaut) pendant la création de la pile. Le délai se produit pendant les mises à jour de la pile uniquement lorsque les propriétés de la ressource personnalisée sont modifiées.

Le modèle provisionne les ressources suivantes :

- une ressource personnalisée,
- une fonction Lambda, et
- un rôle IAM qui permet à Lambda d'écrire des journaux dans CloudWatch

Il définit également deux sorties :

- Le temps réel pendant lequel la fonction a attendu.
- Un identifiant unique généré lors de chaque exécution de la fonction Lambda.

 Note

CloudFormation est un service gratuit, mais Lambda facture en fonction du nombre de requêtes pour vos fonctions et de la durée d'exécution de votre code. Pour plus d'informations sur la tarification Lambda, consultez [Tarification AWS Lambda](#).

Exemple de modèle

Vous pouvez consulter le modèle de ressource personnalisée prise en charge par Lambda avec le mécanisme de délai ci-dessous :

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
          }]
        },
        "Path": "/",
        "Policies": [{
          "PolicyName": "AllowLogs",
          "PolicyDocument": {
            "Statement": [{
              "Effect": "Allow",
              "Action": ["logs:*"],
              "Resource": "*"
            }]
          }
        }]
      }
    },
    "CFNWaiter": {
      "Type": "AWS::Lambda::Function",
```

```

    "Properties": {
      "Handler": "index.handler",
      "Runtime": "python3.9",
      "Timeout": 900,
      "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
      "Code": {
        "ZipFile": { "Fn::Join": ["\n", [
          "from time import sleep",
          "import json",
          "import cfnresponse",
          "import uuid",
          "",
          "def handler(event, context):",
          "  wait_seconds = 0",
          "  id = str(uuid.uuid1())",
          "  if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
          "    wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout", 0))",
          "    sleep(wait_seconds)",
          "    response = {",
          "      \"TimeWaited\": wait_seconds,",
          "      \"Id\": id ",
          "    }",
          "    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,",
          "\"Waiter-\"+id)"
        ]]]
      }
    },
    "CFNWaiterCustomResource": {
      "Type": "AWS::CloudFormation::CustomResource",
      "Properties": {
        "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
        "ServiceTimeout": 60
      }
    },
    "Outputs": {
      "TimeWaited": {
        "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
        "Export": { "Name": "TimeWaited" }
      },
      "WaiterId": {
        "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },

```

```
    "Export": { "Name": "WaiterId" }
  }
}
```

YAML

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |
            from time import sleep
            import json
            import cfnresponse
            import uuid
```

```

def handler(event, context):
    wait_seconds = 0
    id = str(uuid.uuid1())
    if event["RequestType"] in ["Create", "Update"]:
        wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
        sleep(wait_seconds)
    response = {
        "TimeWaited": wait_seconds,
        "Id": id
    }
    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    ServiceTimeout: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId

```

Démonstration du modèle d'exemple

Les extraits suivants expliquent les parties pertinentes du modèle d'exemple afin de vous aider à comprendre comment la fonction Lambda est associée à une ressource personnalisée et à comprendre la sortie.

[AWS::Lambda::Function](#) ressource CFNWaiter

La ressource `AWS::Lambda::Function` spécifie le code source de la fonction, le nom du gestionnaire, l'environnement d'exécution et le rôle d'exécution Amazon Resource Name (ARN).

La propriété `Handler` est définie sur `index.handler` car elle utilise un code source Python.

[Pour plus d'informations sur les identifiants de gestionnaire acceptés lors de l'utilisation de codes source de fonctions en ligne, consultez Code. AWS::Lambda::Function](#)

La propriété `Runtime` est définie sur `python3.9` car le fichier source est un code Python.

La propriété `Timeout` est définie sur 900 secondes.

La propriété `Role` utilise la fonction `Fn::GetAtt` pour obtenir l'ARN du rôle d'exécution `LambdaExecutionRole` déclaré dans la ressource `AWS::IAM::Role` du modèle.

La propriété `Code` définit le code de la fonction en ligne à l'aide d'une fonction Python. La fonction Python du modèle d'exemple effectue les opérations suivantes :

- Créer un ID unique à l'aide de l'UUID
- Vérifier si la demande est une demande de création ou de mise à jour
- Mettre en veille pendant la durée spécifiée pour `ServiceTimeout` pendant les demandes `Create` ou `Update`
- Renvoyer le temps d'attente et l'ID unique

JSON

```
...
"CFNWaiter": {
  "Type": "AWS::Lambda::Function",
  "Properties": {
    "Handler": "index.handler",
    "Runtime": "python3.9",
    "Timeout": 900,
    "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
    "Code": {
      "ZipFile": { "Fn::Join": ["\n", [
        "from time import sleep",
        "import json",
        "import cfnresponse",
        "import uuid",
        "",
        "def handler(event, context):",
        "    wait_seconds = 0",
        "    id = str(uuid.uuid1())",
        "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
        "        wait_seconds = int(event[\"ResourceProperties\"][\"ServiceTimeout\"]),
        "        sleep(wait_seconds)",
        "        response = {",
```



```

        "    \"TimeWaited\": wait_seconds,",
        "    \"Id\": id ",
        "  }",
        "  cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
\"Waiter-\"+id)"
    ]}]
  }
}
},
...

```

YAML

```

...
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile:
        !Sub |
        from time import sleep
        import json
        import cfnresponse
        import uuid

        def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
                wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
                sleep(wait_seconds)
            response = {
                "TimeWaited": wait_seconds,
                "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
...

```

[AWS::IAM::Role](#) ressource LambdaExecutionRole

La ressource `AWS::IAM::Role` crée un rôle d'exécution pour la fonction Lambda, qui inclut une politique d'assomption de rôle permettant à Lambda de l'utiliser. Il contient également une politique autorisant l'accès aux CloudWatch journaux.

JSON

```

...
  "LambdaExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [{
          "Effect": "Allow",
          "Principal": { "Service": ["lambda.amazonaws.com"] },
          "Action": ["sts:AssumeRole"]
        }]
      },
      "Path": "/",
      "Policies": [{
        "PolicyName": "AllowLogs",
        "PolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Action": ["logs:*"],
            "Resource": "*"
          }]
        }
      }]
    }
  },
  ...

```

YAML

```

...
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:

```

```

    - Effect: "Allow"
      Principal:
        Service:
          - "lambda.amazonaws.com"
      Action:
        - "sts:AssumeRole"
    Path: "/"
    Policies:
      - PolicyName: "AllowLogs"
        PolicyDocument:
          Statement:
            - Effect: "Allow"
              Action:
                - "logs:*"
              Resource: "*"
  ...

```

[AWS::CloudFormation::CustomResource](#) ressource CFNWaiterCustomResource

La ressource personnalisée est liée à la fonction Lambda avec son ARN à l'aide de `!GetAtt CFNWaiter.Arn`. Elle mettra en œuvre un temps d'attente de 60 secondes pour les opérations de création et de mise à jour, comme défini dans `ServiceTimeout`. La ressource ne sera invoquée pour une opération de mise à jour que si les propriétés sont modifiées.

JSON

```

...
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  }
},
...

```

YAML

```

...
  CFNWaiterCustomResource:

```

```
Type: AWS::CloudFormation::CustomResource
Properties:
  ServiceToken: !GetAtt CFNWaiter.Arn
  ServiceTimeout: 60
...
```

Outputs

Les Outputs de ce modèle sont les TimeWaited et les WaiterId. La valeur TimeWaited utilise une fonction Fn::GetAtt pour fournir le temps d'attente réel de la ressource. Le WaiterId utilise une fonction Fn::GetAtt pour fournir l'ID unique qui a été généré et associé à l'exécution.

JSON

```
...
"Outputs": {
  "TimeWaited": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
    "Export": { "Name": "TimeWaited" }
  },
  "WaiterId": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
    "Export": { "Name": "WaiterId" }
  }
}
}
```

YAML

```
...
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
```

...

Conditions préalables

Vous devez disposer des autorisations IAM pour utiliser tous les services correspondants, tels que CloudFormation Lambda et.

Lancement de la pile

Pour créer la pile

1. Recherchez le modèle de votre choix (YAML ou JSON) dans la section [Exemple de modèle](#) et enregistrez-le sur votre ordinateur sous le nom `samplelambdabackedcustomresource.template`.
2. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation/>.
3. Depuis la page CloudFormation, choisissez Piles, puis Créer une pile avec de nouvelles ressources (standard).
4. Pour la Condition préalable – Préparer le modèle, sélectionnez Choisir un modèle existant.
5. Pour Spécifier le modèle, sélectionnez Charger un fichier modèle, puis Choisir un fichier.
6. Sélectionnez le fichier de modèle `samplelambdabackedcustomresource.template` que vous avez enregistré précédemment.
7. Choisissez Suivant.
8. Pour Nom de la pile, saisissez **SampleCustomResourceStack** et sélectionnez Suivant.
9. Dans cette procédure détaillée, vous n'avez pas besoin d'ajouter des balises ni de spécifier des paramètres avancés. Dès lors, cliquez sur Suivant.
10. Vérifiez que le nom de la pile est correct, puis sélectionnez Créer.

La création de votre pile peut prendre plusieurs minutes. CloudFormation Pour surveiller la progression, affichez les événements de la pile. Pour de plus amples informations, veuillez consulter [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

Si la création de la pile réussit, toutes les ressources de la pile, telles que la fonction Lambda et la ressource personnalisée, ont été créées. Vous avez utilisé avec succès une fonction Lambda et une ressource personnalisée.

[Si la fonction Lambda renvoie une erreur, consultez les journaux de la fonction dans la console CloudWatch Logs](#). Le nom du flux de journaux est l'ID physique de la ressource personnalisée, qui

se trouve dans les ressources de la pile. Pour plus d'informations, consultez la section [Afficher les données du journal](#) dans le guide de CloudWatch l'utilisateur Amazon.

Nettoyage des ressources

Supprimez la pile pour nettoyer toutes les ressources que vous y avez créées et pour éviter d'être facturé pour des ressources que vous n'utilisez pas.

Pour supprimer la pile

1. Dans la CloudFormation console, choisissez la SampleCustomResourceStackpile.
2. Choisissez Actions, puis Supprimer la pile.
3. Dans le message de confirmation, choisissez Oui, supprimer.

Toutes les ressources que vous avez créées seront supprimées.

Maintenant que vous comprenez comment créer et utiliser une ressource personnalisée prise en charge par Lambda, vous pouvez utiliser le modèle et le code d'exemple de cette démonstration pour créer et tester d'autres piles et fonctions.

Informations connexes

- [CloudFormation Référence sur les ressources personnalisées](#)
- [AWS::CloudFormation::CustomResource](#)

Module **cfn-response**

Dans votre CloudFormation modèle, vous pouvez spécifier une fonction Lambda comme cible d'une ressource personnalisée. Lorsque vous utilisez la propriété `ZipFile` pour définir le code source de votre [fonction](#), vous pouvez charger le module `cfn-response` afin d'envoyer des réponses depuis votre fonction Lambda à une ressource personnalisée. Le module `cfn-response` est une bibliothèque qui simplifie l'envoi de réponses à la ressource personnalisée ayant invoqué votre fonction Lambda. Ce module contient une méthode `send` qui transmet un [objet de réponse](#) à une ressource personnalisée au moyen d'une URL présignée Amazon S3 (appelée `ResponseURL`).

Le module `cfn-response` est disponible uniquement lorsque vous utilisez la propriété `ZipFile` pour écrire votre code source. Il n'est pas disponible pour le code source stocké dans des compartiments Amazon S3. Pour le code stocké dans des compartiments, vous devez écrire vos propres fonctions pour envoyer des réponses.

 Note

Après l'exécution de la méthode `send`, la fonction Lambda s'arrête. Dès lors, tout ce que vous écrivez après cette méthode est ignoré.

Chargement du module **cfn-response**

Pour les fonctions Node.js, utilisez la fonction `require()` pour charger le module `cfn-response`. Par exemple, l'exemple de code suivant crée un objet `cfn-response` avec le nom `response` :

```
var response = require('cfn-response');
```

Pour Python, utilisez l'instruction `import` pour charger le module `cfnresponse`, comme illustré dans l'exemple suivant :

 Note

Utilisez cette instruction `import` exacte. CloudFormation n'inclura pas le module de réponse si vous utilisez d'autres variantes de l'instruction d'importation.

```
import cfnresponse
```

Paramètres de la méthode **send**

Vous pouvez utiliser les paramètres suivants avec la méthode `send`.

`event`

Champs d'une [demande de ressource personnalisée](#).

`context`

Objet propre aux fonctions Lambda qui vous permet de spécifier à quel moment l'exécution de la fonction et des rappels se termine ou pour accéder aux informations à partir de l'environnement d'exécution Lambda. Pour plus d'informations, consultez [Création de fonctions Lambda avec Node.js](#) dans le Guide du développeur AWS Lambda .

responseStatus

Indique si la fonction a été exécutée avec succès. Utilisez les constantes du `cf::response` pour spécifier le statut `SUCCESS` pour les exécutions réussies et `FAILED` pour celles qui ont échoué.

responseData

Champ Data de l'[objet de réponse](#) d'une ressource personnalisée. Les données sont une liste de paires nom-valeur.

physicalResourceId

Facultatif. Identifiant unique de la ressource personnalisée qui appelé la fonction. Par défaut, le module utilise le nom du flux de journal Amazon CloudWatch Logs associé à la fonction Lambda.

La valeur renvoyée pour un `PhysicalResourceId` peut modifier les opérations de mise à jour des ressources personnalisées. Si la valeur renvoyée est la même, on considère qu'il s'agit d'une mise à jour normale. Si la valeur renvoyée est différente, CloudFormation reconnaît la mise à jour comme un remplacement et envoie une demande de suppression à l'ancienne ressource. Pour de plus amples informations, veuillez consulter [AWS::CloudFormation::CustomResource](#).

noEcho

Facultatif. Indique s'il faut masquer la sortie de la ressource personnalisée lorsque celle-ci est récupérée à l'aide de la fonction `Fn::GetAtt`. Si cette valeur est définie sur `true`, toutes les valeurs renvoyées sont masquées par des astérisques (`*****`), à l'exception des informations stockées dans les emplacements spécifiés ci-dessous. Cette valeur est `false` par défaut.

Important

L'utilisation de l'attribut `NoEcho` ne masque aucune information stockée dans les lieux suivants :

- La section des `Metadata` modèles. CloudFormation ne transforme, ne modifie ni n'expédie aucune information que vous incluez dans `Metadata` cette section. Pour de plus amples informations, veuillez consulter [Métadonnées](#).
- La section de modèle `Outputs` Pour de plus amples informations, veuillez consulter [Sorties](#)
- L'attribut `Metadata` d'une définition de ressource. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Nous vous recommandons vivement de ne pas utiliser ces mécanismes pour inclure des informations sensibles, telles que des mots de passe ou des secrets.

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Exemples

Node.js

Dans l'exemple Node.js suivant, la fonction Lambda en ligne multiplie une valeur d'entrée par 5. Les fonctions en ligne sont particulièrement utiles pour les fonctions de petite ampleur, car elles vous permettent de spécifier le code source directement dans le modèle au lieu de créer un package et de l'importer dans un compartiment Amazon S3. Cette fonction utilise la méthode `cfn-response send` pour renvoyer le résultat à la ressource personnalisée qui l'a appelée.

JSON

```
"ZipFile": { "Fn::Join": ["", [  
    "var response = require('cfn-response');",  
    "exports.handler = function(event, context) {",  
    "  var input = parseInt(event.ResourceProperties.Input);",  
    "  var responseData = {Value: input * 5};",  
    "  response.send(event, context, response.SUCCESS, responseData);",  
    "};",  
  ]}] }
```

YAML

```
ZipFile: >  
  var response = require('cfn-response');  
  exports.handler = function(event, context) {  
    var input = parseInt(event.ResourceProperties.Input);  
    var responseData = {Value: input * 5};  
    response.send(event, context, response.SUCCESS, responseData);  
  };
```

Python

Dans l'exemple Python suivant, la fonction Lambda en ligne multiplie un entier par 5.

JSON

```
"ZipFile" : { "Fn::Join" : ["\n", [  
  "import json",  
  "import cfnresponse",  
  "def handler(event, context):",  
  "    responseValue = int(event['ResourceProperties']['Input']) * 5",  
  "    responseData = {}",  
  "    responseData['Data'] = responseValue",  
  "    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,  
    \"CustomResourcePhysicalID\")",  
  ]}] }
```

YAML

```
ZipFile: |  
  import json  
  import cfnresponse  
  def handler(event, context):  
    responseValue = int(event['ResourceProperties']['Input']) * 5  
    responseData = {}  
    responseData['Data'] = responseValue  
    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,  
    "CustomResourcePhysicalID")
```

Code source du module

Rubriques

- [Code source Node.js asynchrone](#)
- [Code source Node.js](#)
- [Code source Python](#)

Code source Node.js asynchrone

Le code suivant représente le module de réponse pour les fonctions Node.js lorsque le gestionnaire est asynchrone. Examinez-le pour comprendre ce que fait le module et pour vous aider à mettre en œuvre vos propres fonctions de réponse.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0
```

```
exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
            Data: responseData
        });

        console.log("Response body:\n", responseBody);

        var https = require("https");
        var url = require("url");

        var parsedUrl = url.parse(event.ResponseURL);
        var options = {
            hostname: parsedUrl.hostname,
            port: 443,
            path: parsedUrl.path,
            method: "PUT",
            headers: {
                "content-type": "",
                "content-length": responseBody.length
            }
        };

        var request = https.request(options, function(response) {
            console.log("Status code: " + parseInt(response.statusCode));
            resolve(context.done());
        });

        request.on("error", function(error) {
```

```

        console.log("send(..) failed executing https.request(..): " +
maskCredentialsAndSignature(error));
        reject(context.done(error));
    });

    request.write(responseBody);
    request.end();
})
}

function maskCredentialsAndSignature(message) {
    return message.replace(/X-Amz-Credential=[^\s]+/i, 'X-Amz-Credential=*****')
        .replace(/X-Amz-Signature=[^\s]+/i, 'X-Amz-Signature=*****');
}

```

Code source Node.js

Le code suivant représente le module de réponse pour les fonctions Node.js lorsque le gestionnaire n'est pas asynchrone. Examinez-le pour comprendre ce que fait le module et pour vous aider à mettre en œuvre vos propres fonctions de réponse.

```

// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream: " + context.logStreamName,
        PhysicalResourceId: physicalResourceId || context.logStreamName,
        StackId: event.StackId,
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        NoEcho: noEcho || false,
        Data: responseData
    });

    console.log("Response body:\n", responseBody);

```

```
var https = require("https");
var url = require("url");

var parsedUrl = url.parse(event.ResponseURL);
var options = {
    hostname: parsedUrl.hostname,
    port: 443,
    path: parsedUrl.path,
    method: "PUT",
    headers: {
        "content-type": "",
        "content-length": responseBody.length
    }
};

var request = https.request(options, function(response) {
    console.log("Status code: " + parseInt(response.statusCode));
    context.done();
});

request.on("error", function(error) {
    console.log("send(..) failed executing https.request(..): " +
maskCredentialsAndSignature(error));
    context.done();
});

request.write(responseBody);
request.end();
}
```

Code source Python

Le code suivant représente le module de réponse pour les fonctions Python :

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0

from __future__ import print_function
import urllib3
import json
import re

SUCCESS = "SUCCESS"
FAILED = "FAILED"
```

```
http = urllib3.PoolManager()

def send(event, context, responseStatus, responseData, physicalResourceId=None,
        noEcho=False, reason=None):
    responseUrl = event['ResponseURL']

    responseBody = {
        'Status' : responseStatus,
        'Reason' : reason or "See the details in CloudWatch Log Stream:
{}".format(context.log_stream_name),
        'PhysicalResourceId' : physicalResourceId or context.log_stream_name,
        'StackId' : event['StackId'],
        'RequestId' : event['RequestId'],
        'LogicalResourceId' : event['LogicalResourceId'],
        'NoEcho' : noEcho,
        'Data' : responseData
    }

    json_responseBody = json.dumps(responseBody)

    print("Response body:")
    print(json_responseBody)

    headers = {
        'content-type' : '',
        'content-length' : str(len(json_responseBody))
    }

    try:
        response = http.request('PUT', responseUrl, headers=headers,
body=json_responseBody)
        print("Status code:", response.status)

    except Exception as e:

        print("send(..) failed executing http.request(..):",
mask_credentials_and_signature(e))

def mask_credentials_and_signature(message):
```

```
message = re.sub(r'X-Amz-Credential=[^\s]+', 'X-Amz-Credential=*****', message,
flags=re.IGNORECASE)
return re.sub(r'X-Amz-Signature=[^\s]+', 'X-Amz-Signature=*****', message,
flags=re.IGNORECASE)
```

Effectuez un traitement personnalisé sur les CloudFormation modèles à l'aide de macros de modèles

Avec les macros, vous pouvez effectuer un traitement personnalisé sur les modèles, qu'il s'agisse d'actions simples telles que find-and-replace des opérations ou de transformations étendues de modèles entiers.

Pour avoir une idée de l'étendue des possibilités, considérez les `AWS::Serverless` transformations `AWS::Include` and, qui sont des macros hébergées par CloudFormation :

- La [transformation AWS::Include](#) vous permet d'insérer des extraits de modèles standard dans vos modèles.
- [AWS::Serverlesstransform](#) prend un modèle entier écrit dans la syntaxe AWS SAM (AWS Serverless Application Model), le transforme et le développe en un CloudFormation modèle conforme. Pour plus d'informations sur les applications sans serveur AWS SAM, consultez le [Guide AWS Serverless Application Model du développeur](#).

Rubriques

- [Facturation](#)
- [Exemples de macros](#)
- [Ressources connexes](#)
- [Vue d'ensemble des CloudFormation macros](#)
- [Création d'une définition de CloudFormation macro](#)
- [Exemple de macro de remplacement de chaîne simple](#)
- [Dépannage du modèle traité](#)

Facturation

Lorsqu'une macro s'exécute, tous les frais liés à l'exécution de la fonction Lambda sont facturés au propriétaire de cette fonction.

Les `AWS::Serverless` transformations `AWS::Include` et sont des macros hébergées par CloudFormation. Aucun frais ne s'applique à leur utilisation.

Exemples de macros

Outre les exemples présentés dans cette section, vous trouverez des exemples de macros, notamment du code source et des modèles, dans notre [GitHub référentiel](#). Ces exemples sont fournis « en l'état » à des fins éducatives.

Ressources connexes

- [AWS::CloudFormation::Macro](#)
- [CloudFormation Transformsection des modèles](#)
- [Fn::Transform](#)
- [Transformation AWS::Serverless](#)
- [Transformation AWS::Include](#)

Vue d'ensemble des CloudFormation macros

Il existe deux étapes principales pour le traitement des modèles à l'aide de macros : la création de la macro elle-même, puis l'utilisation de la macro pour effectuer un traitement sur vos modèles.

Pour créer une définition de macro, vous devez créer les éléments suivants :

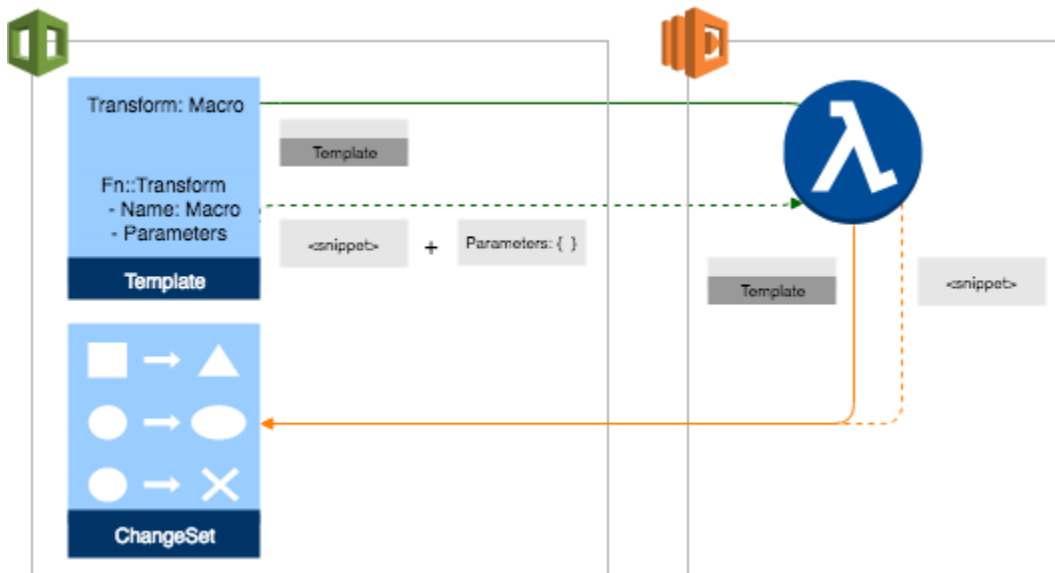
- Une fonction Lambda pour effectuer le traitement du modèle. Cette fonction Lambda accepte un extrait ou un modèle complet, ainsi que tous les paramètres supplémentaires que vous définissez. Elle renvoie l'extrait de modèle traité ou l'ensemble du modèle comme réponse.
- Ressource de type [AWS::CloudFormation::Macro](#), qui permet aux utilisateurs d'appeler la fonction Lambda depuis des modèles. CloudFormation Cette ressource spécifie l'ARN de la fonction Lambda à appeler pour cette macro, ainsi que d'autres propriétés facultatives pour faciliter le débogage. Pour créer cette ressource dans un compte, créez un modèle qui inclut la `AWS::CloudFormation::Macro` ressource, puis créez une pile ou un ensemble de piles avec des autorisations autogérées à partir du modèle. CloudFormation StackSets ne prend actuellement pas en charge la création ou la mise à jour d'ensembles de piles avec des autorisations gérées par les services à partir de modèles faisant référence à des macros.

Pour utiliser une macro, référez la macro dans votre modèle :

- Pour traiter une section ou une partie d'un modèle, référencer la macro dans une fonction `Fn::Transform` située par rapport au contenu du modèle que vous voulez transformer. Lorsque vous utilisez `Fn::Transform`, vous pouvez également transmettre tous les paramètres spécifiés qu'elle requiert.
- Pour traiter un modèle complet, référencez la macro dans la section [Transform](#) du modèle.

En règle générale, vous créez ensuite un jeu de modifications que vous exécutez. (Le traitement des macros peut ajouter plusieurs ressources à votre insu. Pour garantir que vous ayez connaissance de toutes les modifications apportées par les macros, nous vous recommandons vivement d'utiliser des jeux de modifications.) CloudFormation transmet le contenu de modèle spécifié, ainsi que tous les autres paramètres supplémentaires spécifiés, à la fonction Lambda spécifiée dans la ressource de macro. La fonction Lambda renvoie le contenu de modèle traité, qu'il s'agisse d'un extrait ou d'un modèle complet.

Une fois que toutes les macros du modèle ont été appelées, CloudFormation génère un ensemble de modifications qui inclut le contenu du modèle traité. Après avoir passé en revue le jeu de modifications, exécutez-le pour appliquer les modifications.



Comment créer des piles directement

Pour créer ou mettre à jour une pile à l'aide d'un modèle qui référence des macros, en règle générale vous créez un jeu de modifications que vous exécutez ensuite. Un jeu de modifications décrit les actions qu' CloudFormation va entreprendre en fonction du modèle traité. Le traitement des macros peut ajouter plusieurs ressources à votre insu. Pour garantir que vous ayez connaissance de toutes les modifications apportées par les macros, nous vous suggérons vivement d'utiliser des jeux de

modifications. Après avoir passé en revue le jeu de modifications, vous pouvez l'exécuter pour en appliquer les modifications.

Une macro peut ajouter des ressources IAM à votre modèle. Pour ces ressources, CloudFormation vous demande de [reconnaître leurs capacités](#). Comme il est CloudFormation impossible de savoir quelles ressources sont ajoutées avant de traiter votre modèle, vous devrez peut-être reconnaître les fonctionnalités IAM lors de la création de l'ensemble de modifications, selon que les macros référencées contiennent ou non des ressources IAM. Ainsi, lorsque vous exécutez l'ensemble de modifications, il CloudFormation dispose des fonctionnalités nécessaires pour créer des ressources IAM.

Pour créer ou mettre à jour une pile directement à partir d'un modèle traité sans examiner au préalable les modifications proposées dans un jeu de modifications, veuillez spécifier la capacité `CAPABILITY_AUTO_EXPAND` lors d'une demande `CreateStack` ou `UpdateStack`. Ne créez directement de piles à partir d'un modèle de pile qui contient des macros qu'à la condition de connaître les transformations effectuées par la macro. Vous ne pouvez pas utiliser de jeux de modifications avec des macros d'ensemble de piles. Vous devez mettre à jour votre ensemble de piles directement.

Pour plus d'informations, consultez [CreateStack](#) ou [UpdateStack](#) dans la Référence d'API AWS CloudFormation .

Important

Si votre modèle d'ensemble de piles référence une ou plusieurs macros, vous devez créer une pile directement à partir du modèle traité, sans examiner au préalable les modifications qui en résultent dans un jeu de modifications. Le traitement des macros peut ajouter plusieurs ressources à votre insu. Avant de créer ou de mettre à jour un ensemble de piles à partir d'un modèle qui référence directement les macros, assurez-vous de connaître les transformations effectuées par la macro.

Pour réduire le nombre d'étapes de lancement de piles à partir de modèles faisant référence à des macros, vous pouvez utiliser les `deploy AWS CLI` commandes `package` et. Pour plus d'informations, consultez [Téléchargez des artefacts locaux dans un compartiment S3 avec le AWS CLI](#) et [Créer une pile qui inclut des transformations](#).

Considérations

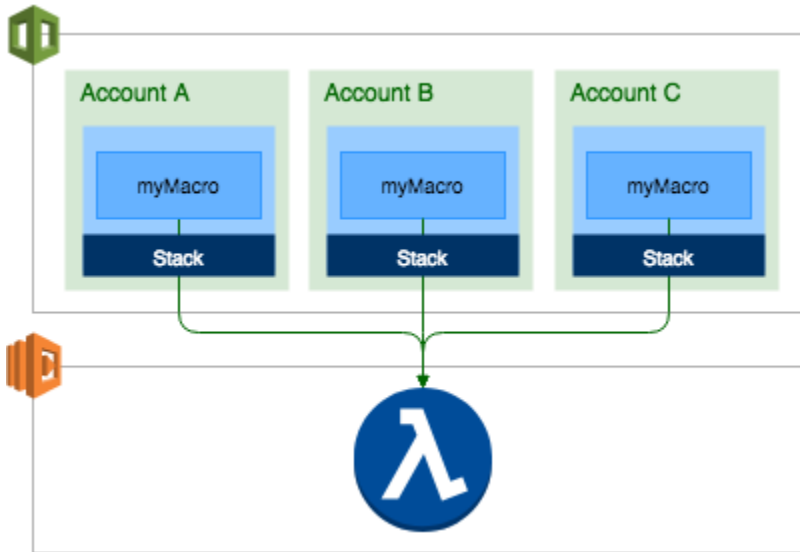
Lorsque vous travaillez avec des macros, veuillez garder à l'esprit les remarques et limitations suivantes :

- Les macros ne sont prises en charge que Régions AWS lorsque Lambda est disponible. Pour obtenir une liste de toutes les régions où Lambda est actuellement disponible, veuillez consulter la rubrique [Points de terminaison et quotas AWS Lambda](#).
- Tout les extraits de modèle traités doivent être dans un format JSON valide.
- Tous les extraits de modèle traités doivent réussir les vérifications de validation pour une opération de création ou de mise à jour de pile ou d'ensemble de piles.
- CloudFormation résout d'abord les macros, puis traite le modèle. Le modèle résultant doit être dans un format JSON valide et ne doit pas dépasser la limite de taille de modèle.
- En raison de l'ordre dans lequel CloudFormation les éléments d'un modèle sont traités, une macro ne peut pas inclure de modules dans le contenu du modèle traité auquel elle renvoie CloudFormation. Pour de plus amples informations, veuillez consulter [Ordre d'évaluation des macros](#).
- Lorsque vous utilisez la fonctionnalité d'annulation des mises à jour, CloudFormation utilise une copie du modèle d'origine. Le modèle d'origine est restauré même si l'extrait inclus a été modifié.
- L'inclusion de macros dans des macros ne fonctionne pas, car nous ne traitons pas les macros de manière récursive.
- La fonction intrinsèque `Fn::ImportValue` n'est pas actuellement prise en charge dans les macros.
- Les fonctions intrinsèques incluses dans le modèle sont évaluées après les macros éventuelles. Par conséquent, le contenu de modèle traité que votre macro renvoie peut inclure des appels à des fonctions intrinsèques, et ils sont évalués comme d'habitude.
- StackSets ne prend actuellement pas en charge la création ou la mise à jour d'ensembles de piles avec des autorisations gérées par les services à partir de modèles faisant référence CloudFormation à des macros.

Portée et autorisations du compte de macro

Vous pouvez utiliser des macros uniquement dans le compte où elles ont été créées en tant que ressource. Le nom de la macro doit être unique au sein d'un compte donné. Toutefois, vous pouvez mettre à disposition les mêmes fonctionnalités dans plusieurs comptes en permettant l'accès entre comptes sur la fonction Lambda, puis en créant des définitions de macro référençant cette fonction

dans plusieurs comptes. Dans l'exemple ci-dessous, trois comptes contiennent des définitions de macro qui pointent toutes vers la même fonction Lambda.



Afin de créer la définition d'une macro, l'utilisateur doit disposer d'autorisations pour créer une pile ou un ensemble de piles dans le compte spécifié.

CloudFormation Pour exécuter correctement une macro incluse dans un modèle, l'utilisateur doit disposer d'Invoke autorisations pour la fonction Lambda sous-jacente. Pour éviter une éventuelle escalade des autorisations, CloudFormation usurpe l'identité de l'utilisateur lors de l'exécution de la macro.

Pour plus d'informations, consultez [Gestion des autorisations dans AWS Lambda](#) du Guide du développeur AWS Lambda et la section [Actions, ressources et clés de condition pour AWS Lambda](#) de la Référence d'autorisation de service.

Création d'une définition de CloudFormation macro

Lorsque vous créez une définition de macro, la définition de macro rend la fonction Lambda sous-jacente disponible dans le compte spécifié afin qu'elle l' CloudFormation invoque pour traiter les modèles.

Mappage d'événements

Lorsqu'elle CloudFormation invoque la fonction Lambda d'une macro, elle envoie une demande au format JSON avec la structure suivante :

```
{
```

```
"region" : "us-east-1",
"accountId" : "$ACCOUNT_ID",
"fragment" : { ... },
"transformId" : "$TRANSFORM_ID",
"params" : { ... },
"requestId" : "$REQUEST_ID",
"templateParameterValues" : { ... }
}
```

- **region**

Région dans laquelle la macro réside.

- **accountId**

ID de compte du compte à partir duquel la macro appelle la fonction Lambda.

- **fragment**

Contenu de modèle disponible pour un traitement personnalisé, au format JSON.

- Pour les macros incluses dans la section Transform du modèle, il s'agit du modèle complet à l'exception de la section Transform.
- Pour les macros incluses dans un appel de fonction intrinsèque `Fn::Transform`, cela inclut tous les nœuds frères (et leurs enfants) en fonction de l'emplacement de la fonction intrinsèque au sein du modèle à l'exception de la fonction `Fn::Transform`. Pour de plus amples informations, veuillez consulter [Portée du modèle de macro](#).

- **transformId**

Nom de la macro qui appelle cette fonction.

- **params**

Pour les appels de `Fn::Transform` fonction, tous les paramètres spécifiés pour la fonction. CloudFormation n'évalue pas ces paramètres avant de les transmettre à la fonction.

Pour les macros incluses dans la section Transform du modèle, cette section est vide.

- **requestId**

ID de la demande qui appelle cette fonction.

- **templateParameterValues**

Tous les paramètres spécifiés dans la [Parameters](#) section du modèle. CloudFormation évalue ces paramètres avant de les transmettre à la fonction.

Format de la réponse

CloudFormation attend de la fonction Lambda qu'elle renvoie une réponse au format JSON suivant :

```
{
  "requestId" : "$REQUEST_ID",
  "status" : "$STATUS",
  "fragment" : { ... },
  "errorMessage": "optional error message for failures"
}
```

- **requestId**

ID de la demande qui appelle cette fonction. Cela doit correspondre à l'ID de demande fourni CloudFormation lors de l'appel de la fonction.

- **status**

Statut de la demande (non sensible à la casse). Doit être réglé sursuccess. CloudFormation traite toute autre réponse comme un échec.

- **fragment**

Le contenu du modèle traité CloudFormation à inclure dans le modèle traité, y compris les frères et sœurs. CloudFormation remplace le contenu du modèle transmis à la fonction Lambda par le fragment de modèle qu'elle reçoit dans la réponse Lambda.

Le contenu de modèle traité doit être dans un format JSON valide et son inclusion dans le modèle traité doit engendrer un modèle valide.

Si votre fonction ne change pas réellement le contenu de modèle qu' CloudFormation lui transmet, mais que vous devez encore inclure ce contenu dans le modèle traité, votre fonction a besoin de renvoyer ce modèle de contenu à CloudFormation , dans sa réponse.

- **errorMessage**

Message d'erreur expliquant pourquoi la transformation a échoué. CloudFormation affiche ce message d'erreur dans le volet Événements de la page d'Informations de la pile pour votre pile.

Par exemple :

```
Error creating change set: Transform
    Compte AWS account
    number::macro name failed with:
    error message string.
```

Créer une définition de macro

Pour créer une définition de CloudFormation macro

1. [Créez une fonction Lambda](#) qui gèrera le traitement du contenu du modèle. Elle peut traiter n'importe quelle partie d'un modèle, jusqu'au modèle entier.
2. Créez un CloudFormation modèle contenant un type de `AWS::CloudFormation::Macro` ressource et spécifiez les `FunctionName` propriétés `Name` et. La `FunctionName` propriété doit contenir l'ARN de la fonction Lambda à invoquer lors de l' CloudFormation exécution de la macro.
3. (Facultatif) Pour faciliter le débogage, vous pouvez également spécifier les propriétés `LogGroupName` et `LogRoleArn` lors de la création du type de ressource `AWS::CloudFormation::Macro` pour votre macro. Ces propriétés vous permettent de spécifier le groupe de CloudWatch journaux auquel CloudFormation envoie les informations de journalisation des erreurs lors de l'appel de la fonction Lambda sous-jacente de la macro, et le CloudFormation rôle doit assumer lors de l'envoi d'entrées de journal à ces journaux.
4. [Créez une pile](#) à l'aide du modèle contenant la macro dans le compte dans lequel vous voulez l'utiliser. Ou bien, [créez un ensemble de piles avec des autorisations autogérées](#) à l'aide du modèle avec la macro dans le compte administrateur, puis créez des instances de pile dans les comptes cibles.
5. Une CloudFormation fois que les piles contenant la définition de la macro ont été créées avec succès, la macro peut être utilisée dans ces comptes. Vous utilisez une macro en la référençant dans le modèle, à l'emplacement approprié correspondant au contenu de modèle que vous souhaitez traiter.

Portée du modèle de macro

Les macros référencées dans la section `Transform` d'un modèle peuvent traiter l'ensemble du contenu de ce modèle.

Les macros référencées dans une fonction `Fn::Transform` peuvent traiter le contenu de tous les éléments frères (y compris de leurs enfants) de cette fonction `Fn::Transform` dans le modèle.

Par exemple, dans l'exemple de modèle ci-dessous, `AWS::Include` peut traiter toutes les propriétés `MyBucket`, en fonction de l'emplacement de la fonction `Fn::Transform` qui les contient. `MyMacro` peut traiter le contenu du modèle complet en raison de son inclusion dans la section `Transform`.

```
# Start of processable content for MyMacro
AWSTemplateFormatVersion: 2010-09-09
Transform: [MyMacro]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    # Start of processable content for AWS::Include
    Properties:
      BucketName: amzn-s3-demo-bucket1
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: 'AWS::Include'
          Parameters:
            Location: s3://amzn-s3-demo-bucket2/MyFileName.yaml
      CorsConfiguration: []
    # End of processable content for AWS::Include
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageID: ami-1234567890abcdef0
# End of processable content for MyMacro
```

Ordre d'évaluation des macros

Vous pouvez référencer plusieurs macros dans un modèle donné, y compris des transformations hébergées par CloudFormation, telles que [AWS::Include](#) et [AWS::Serverless](#).

Les macros sont évaluées dans l'ordre, en fonction de leur emplacement dans le modèle, du plus profondément imbriqué jusqu'au plus général. Les macros figurant dans le même emplacement dans le modèle sont évaluées en série en fonction de l'ordre dans lequel elles sont répertoriées.

Les transformations telles que `AWS::Include` et `AWS::Transform` sont traitées de la même manière que toutes les autres macros en termes d'ordre et de portée d'action.

Par exemple, dans l'exemple de modèle ci-dessous, CloudFormation évalue d'abord la `PolicyAdder` macro, car il s'agit de la macro la plus profondément imbriquée du modèle. CloudFormation évalue ensuite `MyMacro` avant d'évaluer `AWS::Serverless` car il est répertorié précédemment `AWS::Serverless` dans la `Transform` section.

```
AWSTemplateFormatVersion: 2010-09-09
Transform: [MyMacro, AWS::Serverless]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: PolicyAdder
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageID: ami-1234567890abcdef0
```

Exemple de macro de remplacement de chaîne simple

L'exemple suivant passe en revue le processus d'utilisation des macros, de la définition de la macro dans un modèle jusqu'à l'utilisation de la macro dans un modèle, en passant par la création d'une fonction Lambda pour la macro.

Dans cet exemple, nous créons une macro simple qui insère la chaîne spécifiée à la place du contenu cible spécifié dans le modèle traité. Ensuite, nous l'utiliserons pour insérer un champ `WaitHandleCondition` vide dans l'emplacement spécifié dans le modèle traité.

Création d'une macro

Avant d'utiliser une macro, nous devons d'abord effectuer deux tâches : créer la fonction Lambda qui exécute le traitement du modèle souhaité, puis rendre cette fonction Lambda accessible CloudFormation en créant une définition de macro.

L'exemple de modèle suivant contient la définition de notre exemple de macro. Pour rendre la macro disponible dans un format spécifique Compte AWS, créez une pile à partir du modèle. La définition

de macro spécifie le nom de la macro, une brève description et fait référence à l'ARN de la fonction Lambda qui CloudFormation appelle lorsque cette macro est utilisée dans un modèle. (Nous n'avons pas inclus de propriété `LogGroupName` ou `LogRoleARN` pour la journalisation des erreurs.)

Dans cet exemple, supposons que la pile créée à partir de ce modèle soit nommée `JavaMacroFunc`. Comme la propriété `macro Name` est définie sur le nom de la pile, la macro résultante est également nommée `JavaMacroFunc`.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Macro:
    Type: AWS::CloudFormation::Macro
    Properties:
      Name: !Sub '${AWS::StackName}'
      Description: Adds a blank WaitConditionHandle named WaitHandle
      FunctionName: 'arn:aws:lambda:us-east-1:012345678910:function:JavaMacroFunc'
```

Utilisation de la macro

Pour utiliser notre macro, nous l'intégrons dans un modèle à l'aide de la fonction intrinsèque `Fn::Transform`.

Lorsque nous créons une pile à l'aide du modèle ci-dessous, CloudFormation appelle notre exemple de macro. La fonction Lambda sous-jacente remplace une chaîne spécifiée par une autre chaîne spécifiée. Dans ce cas, le résultat est qu'un champ `AWS::CloudFormation::WaitConditionHandle` vide est inséré dans le modèle traité.

```
Parameters:
  ExampleParameter:
    Type: String
    Default: 'SampleMacro'

Resources:
  2a:
    Fn::Transform:
      Name: "JavaMacroFunc"
      Parameters:
        replacement: 'AWS::CloudFormation::WaitConditionHandle'
        target: '$$REPLACEMENT$$'
      Type: '$$REPLACEMENT$$'
```

- La macro à invoquer est spécifiée comme `JavaMacroFunc`, qui provient de l'exemple de définition de macro précédent.
- Deux paramètres sont transmis à cette macro, `target` et `replacement`, qui représentent la chaîne cible et sa valeur de remplacement souhaitée.
- La macro peut fonctionner sur le contenu du nœud `Type`, car `Type` est un frère de la fonction `Fn::Transform` référençant la macro.
- Le champ `AWS::CloudFormation::WaitConditionHandle` obtenu est nommé `2a`.
- Le modèle contient également un paramètre de modèle, `ExampleParameter`, auquel la macro a également accès (mais n'utilise pas dans ce cas).

Données d'entrée Lambda

Lors du CloudFormation traitement de notre exemple de modèle lors de la création de la pile, il transmet le mappage d'événements suivant à la fonction Lambda référencée dans la définition de la `JavaMacroFunc` macro.

- `region` : `us-east-1`
- `accountId` : `012345678910`
- `fragment` :

```
{
  "Type": "$$REPLACEMENT$$"
}
```

- `transformId` : `012345678910::JavaMacroFunc`
- `params` :

```
{
  "replacement": "AWS::CloudFormation::WaitConditionHandle",
  "target": "$$REPLACEMENT$$"
}
```

- `requestId` : `5dba79b5-f117-4de0-9ce4-d40363bfb6ab`
- `templateParameterValues` :

```
{
  "ExampleParameter": "SampleMacro"
}
```

```
}
```

fragment contient du code JSON représentant le fragment de modèle que la macro peut traiter. Ce fragment comprend les frères de l'appel de fonction `Fn::Transform`, mais pas l'appel de fonction lui-même. En outre, `params` contient le code JSON représentant les paramètres de macro. Dans ce cas, `replacement` et `target`. De la même façon, `templateParameterValues` contient du code JSON représentant les paramètres spécifiés pour le modèle dans sa totalité.

Code de la fonction Lambda

Voici le code réel de la fonction Lambda sous-jacente à la macro d'exemple `JavaMacroFunc`. La fonction parcourt le fragment de modèle inclus dans la réponse (qu'elle soit dans un format de chaîne, de liste ou de mappage), à la recherche de la chaîne cible spécifiée. Si elle trouve la chaîne cible spécifiée, la fonction Lambda remplace la chaîne cible par la chaîne de remplacement spécifiée. Sinon, la fonction laisse le fragment de modèle inchangé. La fonction renvoie ensuite une carte des propriétés attendues, décrites en détail ci-dessous, à CloudFormation.

```
package com.macroexample.lambda.demo;

import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class LambdaFunctionHandler implements RequestHandler<Map<String, Object>,
    Map<String, Object>> {

    private static final String REPLACEMENT = "replacement";
    private static final String TARGET = "target";
    private static final String PARAMS = "params";
    private static final String FRAGMENT = "fragment";
    private static final String REQUESTID = "requestId";
    private static final String STATUS = "status";
    private static final String SUCCESS = "SUCCESS";
    private static final String FAILURE = "FAILURE";

    @Override
    public Map<String, Object> handleRequest(Map<String, Object> event, Context
context) {
```

```

    // TODO: implement your handler
    final Map<String, Object> responseMap = new HashMap<String, Object>();
    responseMap.put(REQUESTID, event.get(REQUESTID));
    responseMap.put(STATUS, FAILURE);
    try {
        if (!event.containsKey(PARAMS)) {
            throw new RuntimeException("Params are required");
        }

        final Map<String, Object> params = (Map<String, Object>) event.get(PARAMS);
        if (!params.containsKey(REPLACEMENT) || !params.containsKey(TARGET)) {
            throw new RuntimeException("replacement or target under Params are
required");
        }

        final String replacement = (String) params.get(REPLACEMENT);
        final String target = (String) params.get(TARGET);
        final Object fragment = event.getOrDefault(FRAGMENT, new HashMap<String,
Object>());
        final Object retFragment;
        if (fragment instanceof String) {
            retFragment = iterateAndReplace(replacement, target, (String) fragment);
        } else if (fragment instanceof List) {
            retFragment = iterateAndReplace(replacement, target, (List<Object>) fragment);
        } else if (fragment instanceof Map) {
            retFragment = iterateAndReplace(replacement, target, (Map<String, Object>)
fragment);
        } else {
            retFragment = fragment;
        }
        responseMap.put(STATUS, SUCCESS);
        responseMap.put(FRAGMENT, retFragment);
        return responseMap;
    } catch (Exception e) {
        e.printStackTrace();
        context.getLogger().log(e.getMessage());
        return responseMap;
    }
}

private Map<String, Object> iterateAndReplace(final String replacement, final
String target, final Map<String, Object> fragment) {
    final Map<String, Object> retFragment = new HashMap<String, Object>();
    final List<String> replacementKeys = new ArrayList<>();

```

```

        fragment.forEach((k, v) -> {
            if (v instanceof String) {
                retFragment.put(k, iterateAndReplace(replacement, target, (String)v));
            } else if (v instanceof List) {
                retFragment.put(k, iterateAndReplace(replacement, target, (List<Object>)v));
            } else if (v instanceof Map ) {
                retFragment.put(k, iterateAndReplace(replacement, target, (Map<String, Object>)
v));
            } else {
                retFragment.put(k, v);
            }
        });
        return retFragment;
    }

    private List<Object> iterateAndReplace(final String replacement, final String
target, final List<Object> fragment) {
        final List<Object> retFragment = new ArrayList<>();
        fragment.forEach(o -> {
            if (o instanceof String) {
                retFragment.add(iterateAndReplace(replacement, target, (String) o));
            } else if (o instanceof List) {
                retFragment.add(iterateAndReplace(replacement, target, (List<Object>) o));
            } else if (o instanceof Map) {
                retFragment.add(iterateAndReplace(replacement, target, (Map<String, Object>)
o));
            } else {
                retFragment.add(o);
            }
        });
        return retFragment;
    }

    private String iterateAndReplace(final String replacement, final String target,
final String fragment) {
        System.out.println(replacement + " == " + target + " == " + fragment );
        if (fragment != null AND_AND fragment.equals(target))
            return replacement;
        return fragment;
    }
}

```

Réponse de la fonction Lambda

Voici le mappage auquel la fonction Lambda retourne CloudFormation pour être traitée.

- `requestId` : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- `status` : SUCCESS
- `fragment` :

```
{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}
```

Les `requestId` correspondances envoyées depuis CloudFormation et une `status` valeur de SUCCESS indiquent que la fonction Lambda a traité avec succès le fragment de modèle inclus dans la demande. Dans cette réponse, `fragment` contient le code JSON représentant le contenu à insérer dans le modèle traité à la place de l'extrait de modèle d'origine.

Modèle traité obtenu

Après avoir CloudFormation reçu une réponse positive de la part de la fonction Lambda, elle insère le fragment de modèle renvoyé dans le modèle traité.

Vous trouverez ci-dessous le modèle traité obtenu pour notre exemple. L'appel de fonction intrinsèque `Fn::Transform` qui faisait référence à la macro `JavaMacroFunc` n'est plus inclus. Le fragment de modèle renvoyé par la fonction Lambda est inclus dans l'emplacement approprié, de sorte que le contenu `"Type": "$$REPLACEMENT$$"` a été remplacé par `"Type": "AWS::CloudFormation::WaitConditionHandle"`.

```
{
  "Parameters": {
    "ExampleParameter": {
      "Default": "SampleMacro",
      "Type": "String"
    }
  },
  "Resources": {
    "2a": {
      "Type": "AWS::CloudFormation::WaitConditionHandle"
    }
  }
}
```

```
}  
}
```

Dépannage du modèle traité

Lorsque vous utilisez une macro, le modèle traité se trouve dans la CloudFormation console.

L'étape d'un modèle indique son statut de traitement :

- **Original** : modèle initialement soumis par l'utilisateur pour créer ou mettre à jour la pile ou l'ensemble de piles.
- **Processed**: modèle CloudFormation utilisé pour créer ou mettre à jour la pile ou l'ensemble de piles après le traitement des macros référencées. Le modèle traité est formaté au format JSON, même si le modèle d'origine a été formaté comme YAML.

Pour le dépannage, veuillez utiliser le modèle traité. Si un modèle ne référence pas de macros, le modèle d'origine et le modèle traité sont identiques.

Pour de plus amples informations, veuillez consulter [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

Pour utiliser le AWS CLI pour obtenir le modèle traité, utilisez la [get-template](#) commande.

Limite de taille

La taille maximale d'un modèle de pile traité est de 51 200 octets lorsqu'il est transmis directement dans une demande `CreateStack`, `UpdateStack` ou `ValidateTemplate`, ou de 1 Mo lorsqu'il est transmis en tant qu'objet S3 à l'aide d'une URL de modèle Amazon S3. Cependant, pendant le traitement, l'état temporaire du modèle est mis à CloudFormation jour car il traite en série les macros contenues dans le modèle. De ce fait, la taille du modèle pendant le traitement peut temporairement dépasser la taille autorisée d'un modèle entièrement traité. CloudFormation permet une certaine mémoire tampon pour ces modèles en cours de traitement. Cependant, vous devez concevoir vos modèles et vos macros en gardant à l'esprit la taille maximale autorisée pour un modèle de pile traité.

Si une `Transformation data limit exceeded` erreur est CloudFormation renvoyée lors du traitement de votre modèle, cela signifie que votre modèle a dépassé la taille maximale CloudFormation autorisée pendant le traitement.

Pour résoudre ce problème, envisagez de faire ce qui suit :

- Restructurer votre modèle en plusieurs modèles pour éviter de dépasser la taille maximale autorisée pour les modèles en cours de traitement. Par exemple :
 - Utilisez des modèles de piles imbriquées pour encapsuler des parties du modèle. Pour de plus amples informations, veuillez consulter [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).
 - Créez plusieurs piles et utilisez des références entre piles pour échanger des informations entre elles. Pour de plus amples informations, veuillez consulter [Référez-vous aux sorties de ressources dans une autre CloudFormation pile](#).
- Réduisez la taille du fragment de modèle renvoyé par une macro donnée. CloudFormation n'altère pas le contenu des fragments renvoyés par les macros

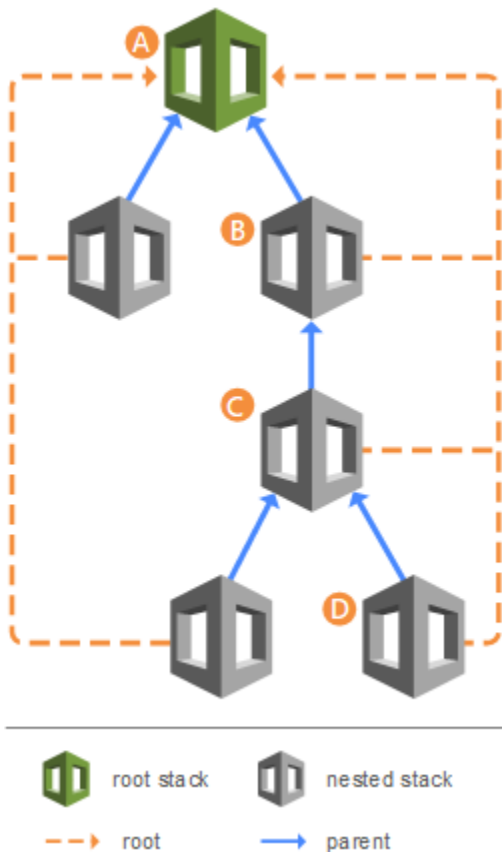
Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées

À mesure que votre infrastructure se développe, vous pouvez être amené à créer de manière répétée des configurations de ressources identiques dans plusieurs modèles. Pour éviter cette redondance, vous pouvez séparer ces configurations communes dans des modèles dédiés. Vous pouvez ensuite utiliser la ressource [AWS::CloudFormation::Stack](#) dans d'autres modèles pour référencer ces modèles dédiés, créant ainsi des piles imbriquées.

Par exemple, supposons que vous disposiez d'une configuration d'équilibreur de charge que vous utilisez pour la plupart de vos piles. Au lieu de copier et coller les mêmes configurations dans vos modèles, vous pouvez créer un modèle dédié pour l'équilibreur de charge. Vous pouvez alors référencer ce modèle à partir d'autres modèles qui nécessitent la même configuration d'équilibreur de charge.

Les piles imbriquées peuvent elles-mêmes contenir d'autres piles imbriquées, ce qui donne une hiérarchie de piles, comme le montre le schéma ci-dessous. La pile racine est la pile de niveau supérieur à laquelle toutes les piles imbriquées appartiennent en fin de compte. Chaque pile imbriquée a une pile parent immédiate. Pour le premier niveau de piles imbriquées, la pile racine est également la pile parent.

- La pile A est la pile racine de toutes les autres piles, imbriquées, dans la hiérarchie.
- Pour la pile B, la pile A est à la fois la pile parent et la pile racine.
- Pour la pile D, la pile C est la pile parent, tandis que pour la pile C, la pile B est la pile parent.



Rubriques

- [Exemple avant et après la division d'un modèle](#)
- [Exemple d'architecture de pile imbriquée](#)
- [Exécution d'opérations sur des piles imbriquées](#)
- [Informations connexes](#)

Exemple avant et après la division d'un modèle

Cet exemple montre comment vous pouvez prendre un CloudFormation modèle unique de grande taille et le réorganiser en un design plus structuré et réutilisable à l'aide de modèles imbriqués. Au départ, le modèle « Avant l'imbrication des piles » affiche toutes les ressources définies dans un seul fichier. Cela peut devenir confus et difficile à gérer à mesure que le nombre de ressources augmente. Le modèle « Après l'imbrication des piles » divise les ressources en modèles plus petits et distincts. Chaque pile imbriquée gère un ensemble spécifique de ressources connexes, ce qui rend la structure globale plus organisée et plus facile à maintenir.

Avant l'imbrication des piles	Après l'imbrication des piles
<pre> AWSTemplateFormatVersion: 2010-09-09 Parameters: InstanceType: Type: String Default: t2.micro Description: The EC2 instance type Environment: Type: String Default: Production Description: The deployment environment Resources: MyEC2Instance: Type: AWS::EC2::Instance Properties: ImageId: ami-1234567890abcdef0 InstanceType: !Ref InstanceType MyS3Bucket: Type: AWS::S3::Bucket </pre>	<pre> AWSTemplateFormatVersion: 2010-09-09 Resources: MyFirstNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /first-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed InstanceType: t3.micro MySecondNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /second-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed Environment: Testing DependsOn: MyFirstNestedStack </pre>

Exemple d'architecture de pile imbriquée

Cette section présente une architecture de pile imbriquée composée d'une pile de niveau supérieur qui fait référence à une pile imbriquée. La pile imbriquée déploie une fonction Lambda Node.js, reçoit une valeur de paramètre de la pile de niveau supérieur et renvoie une sortie qui est exposée via la pile de niveau supérieur.

Rubriques

- [Étape 1 : créez un modèle pour la pile imbriquée sur votre système local](#)
- [Étape 2 : créez un modèle pour la pile de niveau supérieur sur votre système local](#)
- [Étape 3 : packagez et déployez les modèles](#)

Étape 1 : créez un modèle pour la pile imbriquée sur votre système local

L'exemple suivant montre le format du modèle de pile imbriquée.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Nested stack template for Lambda function deployment
Parameters:
  MemorySize:
    Type: Number
    Default: 128
    MinValue: 128
    MaxValue: 10240
    Description: Lambda function memory allocation (128-10240 MB)
Resources:
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Sub "${AWS::StackName}-Function"
      Runtime: nodejs18.x
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          exports.handler = async (event) => {
            return {
              statusCode: 200,
              body: JSON.stringify('Hello from Lambda!')
            };
          };
      MemorySize: !Ref MemorySize
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
```

Outputs:**LambdaArn:**

Description: ARN of the created Lambda function

Value: !GetAtt LambdaFunction.Arn

Étape 2 : créez un modèle pour la pile de niveau supérieur sur votre système local

L'exemple suivant montre le format du modèle de pile de niveau supérieur et la ressource [AWS::CloudFormation::Stack](#) qui fait référence à la pile que vous avez créée à l'étape précédente.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Top-level stack template that deploys a nested stack
Resources:
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: /path_to_template/nested-template.yaml
      Parameters:
        MemorySize: 256
Outputs:
  NestedStackLambdaArn:
    Description: ARN of the Lambda function from nested stack
    Value: !GetAtt NestedStack.Outputs.LambdaArn
```

Étape 3 : packagez et déployez les modèles

 Note

Lorsque vous travaillez avec des modèles localement, la AWS CLI package commande peut vous aider à préparer les modèles pour le déploiement. Elle gère automatiquement le téléchargement des artefacts locaux vers Amazon S3 (y compris TemplateURL) et génère un nouveau fichier modèle avec des références mises à jour vers ces emplacements S3. Pour de plus amples informations, veuillez consulter [Téléchargez des artefacts locaux dans un compartiment S3 avec le AWS CLI](#).

Ensuite, vous pouvez utiliser la commande [package](#) pour charger le modèle imbriqué vers un compartiment Amazon S3.

```
aws cloudformation package \  
  --s3-bucket amzn-s3-demo-bucket \  
  --template /path_to_template/top-level-template.yaml \  
  --output-template-file packaged-template.yaml \  
  --output json
```

La commande génère un nouveau modèle à l'emplacement spécifié par `--output-template-file`. Elle remplace la référence `TemplateURL` par l'emplacement Amazon S3, comme indiqué ci-dessous.

Modèle obtenu

```
AWSTemplateFormatVersion: 2010-09-09  
Description: Top-level stack template that deploys a nested stack  
Resources:  
  NestedStack:  
    Type: AWS::CloudFormation::Stack  
    Properties:  
      TemplateURL: https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket/8b3bb7aa7abfc6e37e2d06b869484bed.template  
      Parameters:  
        MemorySize: 256  
Outputs:  
  NestedStackLambdaArn:  
    Description: ARN of the Lambda function from nested stack  
    Value:  
      Fn::GetAtt:  
        - NestedStack  
        - Outputs.LambdaArn
```

Après avoir exécuté la commande `package`, vous pouvez déployer le modèle traité à l'aide de la commande [deploy](#). Pour les piles imbriquées qui contiennent des ressources IAM, vous devez confirmer les capacités IAM en incluant l'option `--capabilities`.

```
aws cloudformation deploy \  
  --template-file packaged-template.yaml \  
  --stack-name stack-name \  
  --capabilities CAPABILITY_NAMED_IAM
```

Exécution d'opérations sur des piles imbriquées

Lorsque vous travaillez avec des piles imbriquées, vous devez les manipuler avec précaution pendant les opérations. Certaines opérations sur les piles, telles que les mises à jour, doivent être lancées à partir de la pile racine plutôt que d'être effectuées directement sur les piles imbriquées. Lorsque vous mettez à jour une pile racine, seules les piles imbriquées dont le modèle a été modifié seront mises à jour.

De plus, la présence des piles imbriquées peut affecter les opérations sur la pile racine. Par exemple, si une pile imbriquée se bloque dans l'état `UPDATE_ROLLBACK_IN_PROGRESS`, la pile racine attendra que cette pile imbriquée restaure son état avant de continuer. Avant de procéder aux opérations de mise à jour, assurez-vous que vous disposez des autorisations IAM nécessaires pour annuler une mise à jour de pile en cas de restauration. Pour de plus amples informations, veuillez consulter [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

Suivez les procédures suivantes pour trouver la pile racine et les piles imbriquées.

Pour afficher la pile racine d'une pile imbriquée

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Piles, sélectionnez le nom de la pile imbriquée dont vous voulez afficher la pile racine.

Les piles imbriquées affichent NESTED au-dessus de leur nom.

3. Dans l'onglet Infos sur la pile, dans la section Présentation, sélectionnez le nom de la pile répertorié comme Pile racine.

Pour afficher les piles imbriquées qui appartiennent à une pile racine

1. Dans la pile racine dont vous voulez afficher les piles imbriquées, sélectionnez l'onglet Ressources.
2. Dans la colonne Type, recherchez les ressources de type `AWS::CloudFormation::Stack`.

Informations connexes

- [Imbrication d'une pile existante](#)
- [Comprendre les comportements de mise à jour des ressources d'une pile](#)

- [Continuer la restauration à partir des mises à jour de piles imbriquées ayant échoué](#)
- [Échec de la restauration des piles imbriquées](#)

Création de conditions d'attente dans un CloudFormation modèle

Cette rubrique explique comment créer une condition d'attente dans un modèle afin de coordonner la création de ressources de pile ou de suivre la progression d'un processus de configuration. Par exemple, vous pouvez commencer la création d'une autre ressource lorsque la configuration de l'application est partiellement terminée, ou vous pouvez envoyer des signaux lors d'un processus d'installation et de configuration afin de suivre la progression.

When CloudFormation crée une pile qui inclut une condition d'attente :

- Il crée une condition d'attente comme n'importe quelle autre ressource et définit le statut de la condition d'attente sur `CREATE_IN_PROGRESS`.
- CloudFormation attend jusqu'à ce qu'il reçoive le nombre requis de signaux de réussite ou que le délai d'attente ait expiré.
- S'il reçoit le nombre requis de signaux de réussite avant l'expiration du délai d'expiration :
 - Le statut de la condition d'attente passe à `CREATE_COMPLETE`
 - La création de la pile se poursuit
- Si le délai expire ou si un signal d'échec est reçu :
 - Le statut de la condition d'attente passe à `CREATE_FAILED`
 - La pile est restaurée

Important

Pour les ressources Amazon EC2 et Auto Scaling, nous vous recommandons d'utiliser un `CreationPolicy` attribut plutôt que des conditions d'attente. Ajoutez un `CreationPolicy` attribut à ces ressources et utilisez le script d'assistance `cfn-signal` pour signaler qu'un processus de création d'instance s'est terminé avec succès.

Pour plus d'informations, voir la section consacrée à l'[Attribut CreationPolicy](#).

 Note

Si vous en utilisez AWS PrivateLink, les ressources du VPC qui répondent aux conditions d'attente doivent avoir accès à CloudFormation des compartiments Amazon Simple Storage Service (Amazon S3) spécifiques. Les ressources doivent envoyer les réponses aux conditions d'attente à une URL Amazon S3 pré-signée. Si elles ne peuvent pas envoyer des réponses à Amazon S3, CloudFormation ne recevra pas de réponse et l'opération de la pile échouera. Pour plus d'informations, consultez [Accès CloudFormation via un point de terminaison d'interface \(AWS PrivateLink\)](#) et [Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment](#).

Rubriques

- [Création d'une condition d'attente dans votre modèle](#)
- [Syntaxe du signal de condition d'attente](#)
- [Accès aux données du signal](#)

Création d'une condition d'attente dans votre modèle

1. Gestionnaire de condition d'attente

Commencez par définir une ressource [AWS::CloudFormation::WaitConditionHandle](#) dans le modèle de la pile. Cette ressource génère l'URL pré-signée nécessaire à l'envoi de signaux. Cela vous permet d'envoyer un signal sans avoir à fournir vos informations d'identification AWS . Par exemple :

Resources:

MyWaitHandle:

Type: AWS::CloudFormation::WaitConditionHandle

2. Condition d'attente

Ensuite, définissez une ressource [AWS::CloudFormation::WaitCondition](#) dans le modèle de la pile. La structure de base d'un AWS::CloudFormation::WaitCondition se présente comme suit :

MyWaitCondition:

Type: AWS::CloudFormation::WaitCondition

Properties:

```
Handle: String  
Timeout: String  
Count: Integer
```

La ressource `AWS::CloudFormation::WaitCondition` possède deux propriétés obligatoires et une propriété facultative.

- **Handle** (obligatoire) : référence à une ressource `WaitConditionHandle` déclarée dans le modèle.
- **Timeout** (obligatoire) — Le nombre de secondes nécessaires CloudFormation pour attendre la réception du nombre de signaux requis. `Timeout` est une propriété limitée au minimum, ce qui signifie que le délai d'expiration ne survient pas avant l'heure que vous spécifiez, mais qu'il peut survenir peu de temps après. La durée maximale que vous pouvez spécifier est de 43 200 secondes (12 heures).
- **Count** (facultatif) — Le nombre de signaux de réussite que CloudFormation doivent être reçus avant que le statut de cette condition d'attente soit défini sur le statut de cette condition d'attente `CREATE_COMPLETE` et que la création de la pile ne reprenne. Si elle n'est pas spécifiée, la valeur par défaut est 1.

En général, vous voulez que la condition d'attente commence immédiatement après la création d'une ressource spécifique. Pour ce faire, ajoutez l'`DependsOn` attribut à une condition d'attente. Lorsque vous ajoutez un `DependsOn` attribut à une condition d'attente, CloudFormation crée d'abord la ressource dans `DependsOn` l'attribut, puis crée la condition d'attente. Pour plus d'informations, voir la section consacrée à l'[Attribut DependsOn](#).

L'exemple suivant illustre une condition d'attente qui :

- Commence après la création réussie de la ressource `MyEC2Instance`
- Utilise la ressource `MyWaitHandle` comme `WaitConditionHandle`
- A un délai d'expiration de 4500 secondes
- A la valeur `Count` par défaut de 1 (car aucune propriété `Count` n'est spécifiée)

```
MyWaitCondition:  
  Type: AWS::CloudFormation::WaitCondition  
  DependsOn: MyEC2Instance  
  Properties:
```

```
Handle: !Ref MyWaitHandle  
Timeout: '4500'
```

3. Envoi d'un signal

Pour signaler un succès ou un échec CloudFormation, vous exécutez généralement du code ou un script. Par exemple, une application exécutée sur une instance EC2 peut effectuer des tâches de configuration supplémentaires, puis envoyer un signal CloudFormation pour indiquer qu'elles sont terminées.

Le signal doit être envoyé à l'URL pré-signée générée par le descripteur de condition d'attente. Vous utilisez cette URL pré-signée pour signaler la réussite ou l'échec.

Pour envoyer un signal

1. Pour récupérer l'URL pré-signée dans le modèle, utilisez la fonction intrinsèque Ref avec le nom logique du descripteur de condition d'attente.

Comme le montre l'exemple suivant, votre modèle peut déclarer une instance Amazon EC2 et transmettre l'URL pré-signée aux instances EC2 à l'aide de la propriété Amazon EC2 UserData. Cela permet aux scripts ou aux applications exécutés sur ces instances de signaler le succès ou l'échec de CloudFormation.

```
MyEC2Instance:  
  Type: AWS::EC2::Instance  
  Properties:  
    InstanceType: t2.micro # Example instance type  
    ImageId: ami-055e3d4f0bbeb5878 # Change this as needed (Amazon Linux 2023 in  
us-west-2)  
    UserData:  
      Fn::Base64:  
        Fn::Join:  
          - ""  
          - - "SignalURL="  
            - { "Ref": "MyWaitHandle" }
```

Cela donne une sortie UserData similaire à :

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/....
```

Remarque : Dans les outils de ligne de commande AWS Management Console et les outils de ligne de commande, l'URL présignée est affichée sous forme d'identifiant physique de la ressource de gestion des conditions d'attente.

2. (Facultatif) Pour détecter quand la pile entre dans la condition d'attente, vous pouvez utiliser l'une des méthodes suivantes :
 - Si vous créez la pile et que les notifications sont activées, CloudFormation publie une notification à la rubrique spécifiée pour chaque événement de la pile. Si vous ou votre application vous abonnez à cette rubrique, vous pouvez suivre les notifications d'événement de création de descripteur de condition d'attente et récupérer l'URL pré-signée à partir de cette notification.
 - Vous pouvez également surveiller les événements de la pile à l'aide de la AWS Management Console, de l' AWS CLI ou d'un kit SDK.
3. Pour envoyer un signal, vous envoyez un message de requête HTTP à l'aide de l'URL pré-signée. Il doit s'agir d'une méthode de requête PUT, et l'en-tête Content-Type doit être une chaîne vide ou doit être omise. Le message de requête doit correspondre à une structure JSON dont le format est similaire à celui qui est spécifié dans [Syntaxe du signal de condition d'attente](#).

Vous devez envoyer le nombre de signaux de réussite spécifié par la Count propriété afin de CloudFormation poursuivre la création de la pile. Si la valeur de Count est supérieure à 1, la valeur UniqueId de chaque signal doit être unique pour tous les signaux envoyés à une condition d'attente particulière. UniqueId constitue une chaîne alphanumérique arbitraire.

Une commande `curl` est l'un des moyens permettant d'envoyer un signal. L'exemple suivant montre une ligne de commande `curl` qui indique la réussite à une condition d'attente.

```
$ curl -T /tmp/a \
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-
  west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-
  WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
  Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo
  %3D"
```

où le fichier `/tmp/a` contient la structure JSON suivante :

```
{
  "Status" : "SUCCESS",
```

```
"Reason" : "Configuration Complete",  
"UniqueId" : "ID1234",  
"Data" : "Application has completed configuration."  
}
```

Cet exemple montre une ligne de commande `curl` qui envoie le même signal de réussite, sauf qu'il envoie la structure JSON comme paramètre dans la ligne de commande.

```
$ curl -X PUT \  
-H 'Content-Type:' --data-binary '{"Status" : "SUCCESS", "Reason" : "Configuration  
Complete", "UniqueId" : "ID1234", "Data" : "Application has completed  
configuration."}' \  
"https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-  
west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-  
WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?  
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo  
%3D"
```

Syntaxe du signal de condition d'attente

Lorsque vous envoyez des signaux à l'URL générée par le descripteur de condition d'attente, vous devez utiliser le format JSON suivant :

```
{  
  "Status" : "StatusValue",  
  "UniqueId" : "Some UniqueId",  
  "Data" : "Some Data",  
  "Reason" : "Some Reason"  
}
```

Propriétés

Le champ `Status` doit avoir l'une des valeurs suivantes :

- SUCCESS
- FAILURE

Le `UniqueId` champ identifie le signal destiné à CloudFormation. Si la `Count` propriété de la condition d'attente est supérieure à 1, la `UniqueId` valeur doit être unique pour tous les signaux

envoyés pour une condition d'attente particulière ; sinon, le signal CloudFormation sera considéré comme une retransmission du signal précédemment envoyé avec le même signal UniqueId et l'ignorera.

Le champ Data peut contenir toutes les informations que vous voulez renvoyer avec le signal. Vous pouvez accéder à la Data valeur en utilisant la Fn::GetAtt fonction [Fn::](#) dans le modèle.

Le champ Reason est une chaîne de caractères dont le contenu n'est soumis à aucune autre restriction que la conformité JSON.

Accès aux données du signal

Pour accéder aux données envoyées par des signaux valides, vous pouvez créer une valeur de sortie pour la condition d'attente dans votre CloudFormation modèle. Par exemple :

Outputs:

WaitConditionData:

Description: *The data passed back as part of signalling the WaitCondition*

Value: !GetAtt *MyWaitCondition*.Data

Vous pouvez ensuite consulter ces données à l'aide de la [describe-stacks](#) commande ou de l'onglet Sorties de la CloudFormation console.

La Fn::GetAtt fonction renvoie le UniqueId et Data sous forme de name/value paire au sein d'une structure JSON. Par exemple :

```
{"Signal1":"Application has completed configuration."}
```

Créez des configurations de ressources réutilisables qui peuvent être incluses dans les modèles à l'aide de CloudFormation modules

Les modules vous permettent d'empaqueter des configurations de ressources pour les inclure dans les modèles de pile, de manière transparente, gérable et reproductible. Les modules peuvent encapsuler les configurations de service courantes et les bonnes pratiques en tant que blocs de création modulaires et personnalisables à inclure dans vos modèles de pile. Les modules vous permettent d'inclure des configurations de ressources qui intègrent les bonnes pratiques, les connaissances des experts du secteur et les directives acceptées (pour des domaines tels que la sécurité, la conformité, la gouvernance et les réglementations sectorielles) dans vos modèles, sans avoir à acquérir une connaissance approfondie des subtilités de l'implémentation des ressources.

Par exemple, un expert du domaine des réseaux pourrait créer un module contenant des groupes de sécurité intégrés et des ingress/egress règles conformes aux directives de sécurité. Vous pouvez ensuite inclure ce module dans votre modèle pour mettre en place une infrastructure réseau sécurisée dans votre pile, sans avoir à perdre du temps à comprendre le fonctionnement des sous-réseaux VPCs, des groupes de sécurité et des passerelles. Et comme les modules sont versionnés, si les directives de sécurité changent au fil du temps, l'auteur du module peut créer une nouvelle version du module intégrant ces modifications.

Les caractéristiques de l'utilisation des modules dans vos modèles sont les suivantes :

- **Prévisibilité** — Un module doit respecter le schéma qu'il enregistre dans le CloudFormation registre, afin que vous sachiez quelles ressources il peut utiliser une fois que vous l'avez inclus dans votre modèle.
- **Réutilisabilité** : vous pouvez utiliser le même module sur plusieurs modèles et comptes.
- **Traçabilité** : CloudFormation conserve la connaissance des ressources d'une pile qui ont été provisionnées à partir d'un module, ce qui vous permet de comprendre facilement la source des modifications apportées aux ressources.
- **Facilité de gestion** — Une fois que vous avez enregistré un module, vous pouvez le gérer via le CloudFormation registre, y compris le contrôle des versions et la disponibilité des comptes et des régions.

Un module peut contenir :

- une ou plusieurs ressources à provisionner à partir du module, ainsi que toutes les données associées, telles que les sorties ou les conditions ;
- tous les paramètres du module, qui vous permettent de spécifier des valeurs personnalisées chaque fois que le module est utilisé.

Pour plus d'informations sur le développement de modules, consultez la section [Développement de modules](#) dans le guide de l'utilisateur de la CloudFormation CLI.

Rubriques

- [Considérations relatives à l'utilisation de modules](#)
- [Compréhension de la gestion des versions des modules](#)
- [Utiliser les modules du registre CloudFormation privé](#)
- [Utilisez des paramètres pour spécifier les valeurs des modules](#)

- [Ressources du module de référence dans les CloudFormation modèles](#)

Considérations relatives à l'utilisation de modules

- L'utilisation d'un module n'implique aucun coût supplémentaire. Vous ne payez que pour les ressources que ces modules résolvent dans vos piles.
- CloudFormation les quotas, tels que le nombre maximum de ressources autorisées dans une pile ou la taille maximale du corps du modèle, s'appliquent au modèle traité, que les ressources incluses dans ce modèle proviennent de modules ou non. Pour de plus amples informations, veuillez consulter [Comprendre les CloudFormation quotas](#).
- Les balises que vous spécifiez au niveau de la pile sont affectées aux ressources individuelles dérivées du module.
- Les scripts d'assistance spécifiés au niveau du module ne se propagent pas aux ressources individuelles contenues dans le module lors du traitement du CloudFormation modèle.
- Les sorties spécifiées dans le module sont propagées aux sorties au niveau du modèle.

Chaque sortie se voit attribuer un ID logique qui est une concaténation du nom logique du module et du nom de sortie tel que défini dans le module. Pour de plus amples informations, veuillez consulter [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#).

- Les paramètres spécifiés dans le module ne sont pas propagés aux paramètres au niveau du modèle.

Toutefois, vous pouvez créer des paramètres au niveau du modèle qui font référence aux paramètres au niveau du module. Pour de plus amples informations, veuillez consulter [Utilisez des paramètres pour spécifier les valeurs des modules](#).

Compréhension de la gestion des versions des modules

Le CloudFormation registre agit comme un référentiel dans lequel vous pouvez enregistrer et gérer des modules à utiliser au sein de votre région Compte AWS et de votre région. Vous pouvez enregistrer des modules provenant de différentes sources AWS, y compris des éditeurs tiers, et vos propres extensions personnalisées, dans votre compte et dans votre région. Pour de plus amples informations, veuillez consulter [Gestion des extensions avec le CloudFormation registre](#).

Les modules peuvent avoir différentes versions, vous pouvez donc spécifier la version d'un module que vous souhaitez utiliser. Cette fonctionnalité de gestion des versions est particulièrement utile

lorsque vous devez mettre à jour ou modifier un module sans perturber les piles existantes qui en dépendent.

Gardez les considérations suivantes à l'esprit lorsque vous utilisez plusieurs versions d'un module :

- Pendant les opérations de pile, CloudFormation utilise la version du module actuellement enregistrée comme version par défaut dans la région Compte AWS et dans laquelle l'opération de pile est effectuée. Cela inclut les modules imbriqués dans d'autres modules.

Par conséquent, sachez que si différentes versions du même module sont enregistrées comme version par défaut dans différents comptes ou régions, l'utilisation du même modèle peut entraîner des résultats différents.

- Pendant les opérations de pile, CloudFormation utilise la version de la ressource actuellement enregistrée comme version par défaut dans la région Compte AWS et dans laquelle l'opération de pile est effectuée. Cela inclut les ressources générées par l'inclusion de modules.
- La modification de la version par défaut d'un module ne lance aucune opération de mise à jour de la pile. Toutefois, la prochaine fois que vous effectuerez une opération de pile avec un modèle contenant ce module, comme une mise à jour de pile, la nouvelle version par défaut CloudFormation sera utilisée dans l'opération.

La seule exception est d'effectuer une mise à jour de pile avec l'option spécifiée Utiliser le modèle précédent, comme décrit ci-dessous.

- Pour les opérations de mise à jour de pile, si vous spécifiez l'option Utiliser le CloudFormation modèle précédent, utilisez le modèle traité précédent pour la mise à jour de pile et ne retirez pas le module pour les modifications que vous pourriez y avoir apportées.
- Pour garantir des résultats uniformes, si vous incluez des modules dans un modèle de pile à utiliser avec des ensembles de pile, vous devez vous assurer que la même version du module est configurée de la même manière que la version par défaut dans tous les comptes et régions dans lesquels vous prévoyez de déployer vos instances de pile. Cela inclut les modules imbriqués dans d'autres modules. Pour de plus amples informations, veuillez consulter [Gérez les piles entre les comptes et les régions avec StackSets](#).

Conditions requises pour activer des modules publics tiers

Pour activer avec succès un module public tiers dans votre compte et votre région, les conditions suivantes doivent être remplies pour chaque extension publique tierce (ressource ou module) incluse dans le module :

- Activation de l'extension : l'extension doit être activée dans le compte et dans la région dans lesquels vous souhaitez l'utiliser. Pour de plus amples informations, veuillez consulter [Utiliser des extensions publiques tierces depuis le CloudFormation registre](#).
- Enregistrement d'alias : si l'extension du module utilise un alias de nom de type, l'extension doit être enregistrée dans votre compte et votre région en utilisant le même alias de nom de type. Pour de plus amples informations, veuillez consulter [Utilisez des alias pour faire référence aux extensions](#).
- Compatibilité des versions : la version d'extension actuellement activée doit être l'une des versions majeures prises en charge de cette extension spécifiée dans le module.

Si les extensions et versions d'extension publiques tierces ne sont pas activées correctement, l'opération CloudFormation échouera et une erreur indiquera les extensions et les versions qui doivent être activées pour que le module puisse être activé avec succès.

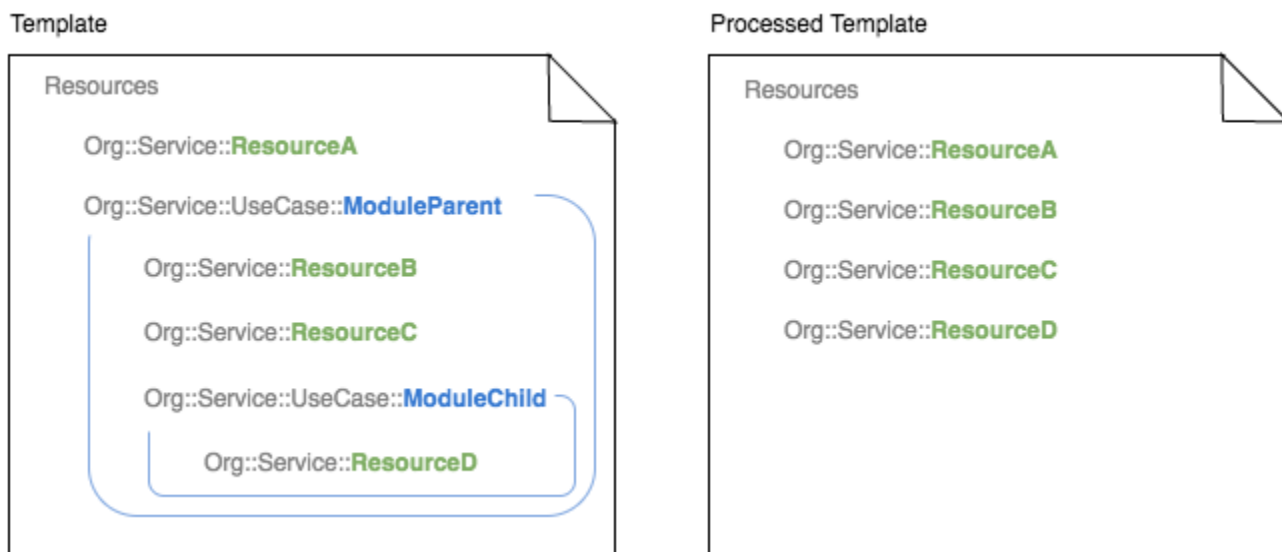
Utiliser les modules du registre CloudFormation privé

Cette rubrique explique comment utiliser les modules dans les CloudFormation modèles. Considérez les modules comme des ensembles de ressources prédéfinis que vous pouvez ajouter à vos modèles.

Pour utiliser un module, procédez comme suit :

- Enregistrer le module — Vous enregistrez les modules dans le CloudFormation registre en tant qu'extensions privées. Assurez-vous qu'il est enregistré dans la région Compte AWS et dans laquelle vous travaillez. Pour de plus amples informations, veuillez consulter [CloudFormation concepts de registre](#).
- Incluez-le dans votre modèle : ajoutez le module à la [Resources](#) section de votre CloudFormation modèle, comme vous le feriez avec d'autres ressources. Vous devrez également fournir toutes les propriétés requises pour le module.
- Création ou mise à jour de la pile : lorsque vous lancez une opération de pile, CloudFormation génère un modèle traité qui transforme tous les modules inclus en ressources appropriées.
- Aperçu des modifications : avant d'apporter des modifications, vous pouvez utiliser un ensemble de modifications pour voir quelles ressources seront ajoutées ou modifiées. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

Prenons l'exemple suivant : Vous disposez d'un modèle qui contient à la fois des ressources et des modules. Le modèle contient une ressource individuelle, `ResourceA`, ainsi qu'un module, `ModuleParent`. Ce module contient deux ressources, `ResourceB` et `ResourceC`, ainsi qu'un module imbriqué, `ModuleChild`. `ModuleChild` contient une seule ressource, `ResourceD`. Si vous créez une pile à partir de ce modèle, CloudFormation traite le modèle et associe les modules aux ressources appropriées. La pile obtenue comporte quatre ressources : `ResourceA`, `ResourceB`, `ResourceC`, et `ResourceD`.



CloudFormation permet de savoir quelles ressources d'une pile ont été créées à partir de modules. Vous pouvez afficher ces informations dans les onglets Événements, Ressources et Écarts pour une pile donnée. Elles sont également comprises dans les prévisualisations des ensembles de modifications.

Les modules se distinguent des ressources d'un modèle, car ils adhèrent à la convention de dénomination en quatre parties suivante, par opposition à la convention typique en trois parties utilisée par les ressources :

```
organization::service::use-case::MODULE
```

Utilisez des paramètres pour spécifier les valeurs des modules

Dans CloudFormation, vous pouvez utiliser les paramètres du modèle pour personnaliser vos piles en fournissant des valeurs d'entrée lors de la création ou de la mise à jour des piles. Ces paramètres vous permettent de modifier certains aspects de la pile en fonction de vos besoins. Pour plus d'informations sur la définition des paramètres de modèle, consultez [CloudFormation Parameters syntaxe du modèle](#).

De même, les modules peuvent également avoir des paramètres. Ces paramètres de module vous permettent d'entrer des valeurs personnalisées dans le module à partir du modèle (ou d'un autre module) qui l'utilise. Le module peut ensuite utiliser ces valeurs personnalisées pour définir les valeurs de propriété des ressources qu'il contient.

Vous pouvez également définir des paramètres de modèle qui définissent les propriétés du module, afin de pouvoir saisir des valeurs qui seront transmises au module au moment de l'opération de pile.

Si un module contient un module imbriqué qui possède ses propres paramètres de module, vous pouvez soit :

- spécifier les valeurs des paramètres du module imbriqué directement dans le module parent ;
- définir les paramètres correspondants du module dans le module parent qui permettent de configurer les paramètres du module imbriqué avec le modèle (ou le module) dans lequel le module parent est contenu.

Utilisation des paramètres de modèle pour spécifier les valeurs des paramètres de module

L'exemple suivant montre comment définir des paramètres de modèle qui transmettent des valeurs à un module.

Ici, le modèle contenant `My::S3::SampleBucket::MODULE` définit un paramètre de modèle, `BucketName`, qui permet à l'utilisateur de spécifier un nom de compartiment S3 pendant l'opération de pile.

```
# Template containing My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucket::MODULE'
    Properties:
      BucketName: !Ref BucketName
```

Spécification des propriétés sur les ressources d'un module enfant à partir du module parent

L'exemple suivant illustre comment spécifier des valeurs de paramètre dans un module imbriqué dans un autre module.

Ce premier module, `My::S3::SampleBucketPrivate::MODULE`, sera le module enfant. Il définit deux paramètres : `BucketName` et `AccessControl`. Les valeurs spécifiées pour ces paramètres permettent de spécifier les propriétés `BucketName` et `AccessControl` de la ressource `AWS::S3::Bucket` contenue par le module. Ci-dessous se trouve le fragment de modèle pour `My::S3::SampleBucketPrivate::MODULE`.

```
# My::S3::SampleBucketPrivate::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket with Versioning and DeletionPolicy.
Parameters:
  BucketName:
    Description: Name for the bucket
    Type: String
  AccessControl:
    Description: AccessControl for the bucket
    Type: String
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref BucketName
      AccessControl: !Ref AccessControl
      DeletionPolicy: Retain
      VersioningConfiguration:
        Status: Enabled
```

Ensuite, le module précédent est imbriqué dans un module parent,

`My::S3::SampleBucket::MODULE`. Le module parent, `My::S3::SampleBucket::MODULE`, définit les paramètres du module enfant de la manière suivante :

- Il définit le paramètre `AccessControl` de `My::S3::SampleBucketPrivate::MODULE` à `Private`.
- Pour `BucketName`, il définit un paramètre de module qui permettra de spécifier le nom du compartiment dans le modèle (ou le module) qui contient `My::S3::SampleBucket::MODULE`.

```
# My::S3::SampleBucket::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket. With Private AccessControl.
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucketPrivate::MODULE'
    Properties:
      BucketName: !Ref BucketName
      AccessControl: Private
```

Spécification des contraintes pour les paramètres de module

Les paramètres du module ne prennent pas en charge l'application de Type ou Contrainte. Pour effectuer une vérification des contraintes sur un paramètre de module, créez un paramètre de modèle avec les contraintes souhaitées. Référez ensuite ce paramètre de modèle dans le paramètre de votre module. Pour plus d'informations sur la définition des paramètres de modèle, consultez [CloudFormation Parameterssyntaxe du modèle](#).

Ressources du module de référence dans les CloudFormation modèles

Dans les CloudFormation modèles, vous devez souvent définir les propriétés d'une ressource en fonction du nom ou de la propriété d'une autre ressource. Pour de plus amples informations, veuillez consulter [Référencement des ressources](#).

Pour référencer une ressource contenue dans un module de votre CloudFormation modèle, vous devez combiner deux noms logiques :

- Le nom logique que vous avez donné au module lui-même lorsque vous l'avez inclus dans votre modèle.
- Le nom logique de la ressource spécifique de ce module.

Vous pouvez combiner ces deux noms logiques avec ou sans point (.) entre eux. Par exemple, si le nom logique du module est `MyModule` et que le nom logique de la ressource est `MyBucket`, vous pouvez faire référence à cette ressource sous le nom `MyModule.MyBucket` ou `MyModuleMyBucket`.

Pour trouver les noms logiques des ressources à l'intérieur d'un module, vous pouvez consulter le schéma du module, qui est disponible dans le CloudFormation registre ou en utilisant l'[DescribeType](#) opération. Le schéma répertorie toutes les ressources et leurs noms logiques qui font partie du module.

Une fois que vous avez le nom logique complet, vous pouvez utiliser des CloudFormation fonctions telles que `GetAtt` et `Ref` pour accéder aux valeurs des propriétés sur les ressources du module.

Par exemple, vous avez un module `My::S3::SampleBucket::MODULE` qui contient une ressource `AWS::S3::Bucket` avec le nom logique `S3Bucket`. Pour faire référence au nom de ce compartiment à l'aide de la fonction `Ref`, vous combinez le nom du module dans votre modèle (`MyBucket`) avec le nom logique de la ressource dans le module (`S3Bucket`). Le nom logique complet est `MyBucket.S3Bucket` ou `MyBucketS3Bucket`.

Exemple de modèle

Le modèle suivant crée un compartiment S3 à l'aide du module `My::S3::SampleBucket::MODULE`. Il crée également une file d'attente Amazon SQS et lui attribue le même nom que le nom du compartiment du module. De plus, le modèle génère l'Amazon Resource Name (ARN) du compartiment S3 créé.

```
# Template that uses My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: My::S3::SampleBucket::MODULE
    Properties:
      BucketName: !Ref BucketName
  exampleQueue:
    Type: AWS::SQS::Queue
    Properties:
      QueueName: !Ref MyBucket.S3Bucket
Outputs:
  BucketArn:
    Value: !GetAtt MyBucket.S3Bucket.Arn
```

Gérer AWS les ressources en tant qu'unité unique avec des CloudFormation piles

Une pile est un ensemble de AWS ressources que vous pouvez gérer comme une seule unité. En d'autres termes, vous pouvez créer, mettre à jour et supprimer un ensemble de ressources en créant, en mettant à jour et en supprimant des piles.

La création d'une pile implique le déploiement d'un CloudFormation modèle qui spécifie les ressources et leurs configurations, CloudFormation puis les provisionne et les configure.

La mise à jour d'une pile implique de modifier le modèle ou les paramètres. CloudFormation compare les modifications que vous soumettez avec l'état actuel de votre pile et met à jour uniquement les ressources modifiées. CloudFormation peut interrompre des ressources ou remplacer des ressources mises à jour, selon les propriétés que vous mettez à jour. Pour plus d'informations sur les mises à jour de ressources, consultez [Comprendre les comportements de mise à jour des ressources d'une pile](#).

CloudFormation propose deux méthodes pour mettre à jour les piles :

- **Ensembles de modifications** : avec les ensembles de modifications, vous pouvez prévisualiser les modifications que CloudFormation seront apportées à votre pile, puis décider d'appliquer ou non ces modifications. Les ensembles de modifications sont des documents au format JSON qui résument les modifications apportées CloudFormation à une pile. Utilisez des ensembles de modifications lorsque vous voulez vous assurer que CloudFormation cela n'entraîne pas de modifications involontaires ou lorsque vous souhaitez envisager plusieurs options. Par exemple, vous pouvez utiliser un ensemble de modifications pour vérifier qu'il CloudFormation ne remplacera pas les instances de base de données de votre pile lors d'une mise à jour.
- **Mise à jour directe** : lorsque vous mettez à jour directement une pile, vous soumettez des modifications et vous les déployez CloudFormation immédiatement. Utilisez les mises à jour directes lorsque vous souhaitez déployer des mises à jour rapidement.

La suppression d'une pile entraîne la suppression des ressources qui lui sont associées. Par exemple, une pile peut inclure toutes les ressources requises pour exécuter une application web, telle qu'un serveur web, une base de données et des règles de mise en réseau. Si vous n'avez plus besoin de cette application web, vous pouvez simplement supprimer la pile, et toutes les ressources associées seront supprimées.

 Note

Des frais vous seront facturés pour toute la durée pendant laquelle les ressources de la pile ont été exécutées (même si vous avez supprimé la pile tout de suite).

CloudFormation garantit que toutes les ressources de la pile sont créées ou supprimées comme il convient. Comme CloudFormation traite les ressources de la pile en tant qu'unité unique, elles doivent toutes être créées ou supprimées pour que la pile soit créée ou supprimée. Si la création d'une ressource échoue, CloudFormation restaure la pile et supprime automatiquement les ressources qui ont été créées. Si la suppression d'une ressource échoue, toutes les ressources restantes sont conservées tant que la pile n'est pas supprimée.

Rubriques

- [Interfaces pour gérer vos piles](#)
- [Création d'une pile depuis la CloudFormation console](#)
- [Afficher les informations relatives à la pile depuis la CloudFormation console](#)
- [Mettre à jour votre modèle de pile](#)
- [Comprendre les comportements de mise à jour des ressources d'une pile](#)
- [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#)
- [Valider les déploiements de stack](#)
- [Mettre à jour directement les piles](#)
- [Annuler la mise à jour d'une pile](#)
- [Supprimer une pile de la CloudFormation console](#)
- [Surveiller la progression de la pile](#)
- [Réduisez votre batterie en CloudFormation cas de violation d'alarme grâce à des déclencheurs](#)
- [Déectez les modifications de configuration non gérées apportées aux piles et aux ressources grâce à la détection des écarts](#)
- [Importer AWS des ressources dans une CloudFormation pile](#)
- [Refactorisation de piles](#)
- [Prise en charge des types de ressources](#)
- [Utilisez des liens de création rapide pour créer des piles CloudFormation](#)

- [Exemples de commandes d'opération de CloudFormation pile pour AWS CLI et PowerShell](#)

Interfaces pour gérer vos piles

Vous pouvez gérer vos CloudFormation piles à l'aide des interfaces suivantes :

- CloudFormation console — Fournit une interface Web que vous pouvez utiliser pour accéder à vos piles. Vous pouvez accéder à la CloudFormation console en vous connectant au AWS Management Console, en utilisant le champ de recherche de la barre de navigation pour effectuer une recherche CloudFormation, puis en choisissant CloudFormation parmi les résultats de recherche.
- AWS Command Line Interface— Fournit des commandes pour un large éventail de AWS services CloudFormation, notamment, et est compatible avec Windows, Mac et Linux. Pour plus d'informations sur les CloudFormation commandes, consultez [cloudformation](#) dans la référence des AWS CLI commandes.
- Outils AWS pour PowerShell : un ensemble de modules PowerShell qui s'appuient sur les fonctionnalités exposées par le SDK pour .NET. Les outils vous PowerShell permettent de scripter des opérations sur vos AWS ressources à partir de la ligne de PowerShell commande. [Vous trouverez les applets de commande correspondants CloudFormation dans la référence des applets de Outils AWS pour PowerShell commande.](#)
- API de requête : Fournit des actions d'API de bas niveau appelées à l'aide de demandes HTTPS. Si vous effectuez des appels API dans votre application, vous devez écrire le code permettant de gérer les détails de bas niveau, tels que la génération du hachage pour signer la requête. Pour plus d'informations sur les actions d'API pour CloudFormation, consultez la section [Actions](#) de la référence AWS CloudFormation d'API.
- AWS SDKs— Fournit des informations spécifiques à la langue APIs et prend en charge de nombreux détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Pour plus d'informations, consultez [Outils pour développer des outils AWS](#).
- AWS Cloud Development Kit (AWS CDK)— AWS CDK Il s'agit d'un framework de développement logiciel open source qui vous permet de définir une AWS infrastructure à l'aide de langages de programmation familiers tels que Python TypeScript, Java et .NET. Avec le CDK, vous pouvez modéliser les ressources de votre application, puis les approvisionner CloudFormation directement depuis votre environnement de développement intégré (IDE). Pour de plus amples informations, veuillez consulter [AWS Cloud Development Kit \(AWS CDK\)](#).

Création d'une pile depuis la CloudFormation console

Vous pouvez créer un modèle de pile, puis l'utiliser pour créer une pile à l'aide de la CloudFormation console ou d'un outil de ligne de commande. La console propose une interface guidée avec des options prédéfinies, ce qui simplifie la création de la pile.

Rubriques

- [Création d'une pile](#)
- [Configuration des options de pile](#)
- [Prévisualisation de la configuration de la pile](#)

Création d'une pile

Suivez les étapes de cette section pour déployer votre modèle et créer une pile.

Conditions préalables

- Vous devez avoir préalablement créé un modèle de pile. Pour de plus amples informations, veuillez consulter [Utilisation de CloudFormation modèles](#).

Création d'une pile (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le Région AWS pour créer le stack in.
3. Sur la page Piles, choisissez Créer une pile, puis Avec de nouvelles ressources (standard).

Vous pouvez également choisir l'option Avec les ressources existantes (importer des ressources) pour importer les AWS ressources existantes décrites dans votre modèle. Pour plus d'informations sur cette option, consultez [Importer AWS des ressources dans une CloudFormation pile](#).

4. Sur la page Créer une pile, procédez de la manière suivante :
 - Pour utiliser un modèle existant, dans Prérequis - Préparer le modèle, sélectionnez Choisir un modèle existant. Ensuite, sous Spécifier le modèle, choisissez soit URL Amazon S3, soit Charger un modèle de fichier, en fonction de l'emplacement du modèle.
 - Si vous choisissez URL Amazon S3, fournissez une URL vers le modèle de fichier dans un compartiment S3.

Si votre modèle comporte des piles imbriquées (par exemple des piles décrites dans d'autres fichiers de modèle situés dans des sous-répertoires), assurez-vous que votre compartiment S3 contient les fichiers et répertoires nécessaires.

Si vous avez un modèle provenant d'un compartiment avec la gestion des versions activée, vous pouvez spécifier une version particulière du modèle en ajoutant ?
versionId=*version-id* à l'URL. Pour plus d'informations sur les compartiments avec la gestion des versions activée, consultez [Travail avec des objets dans un compartiment avec gestion des versions activée](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

L'URL doit pointer vers un modèle d'une taille maximale de 1 Mo, stocké dans un compartiment S3 pour lequel vous disposez d'autorisations de lecture. L'URL peut avoir une longueur de 1 024 caractères au maximum. Certaines ressources peuvent nécessiter que le compartiment se situe dans la même région que la pile.

- Si vous choisissez Charger un modèle de fichier, choisissez Choisir un fichier pour choisir un modèle de fichier sur votre ordinateur local. La taille du modèle de fichier doit être inférieure ou égale à 1 Mo.

Une fois que vous avez choisi votre modèle, CloudFormation télécharge le fichier et affiche l'URL S3. CloudFormation le télécharge dans un compartiment Amazon S3 de votre Compte AWS. Si vous possédez déjà un compartiment S3 créé par CloudFormation in your Compte AWS, CloudFormation ajoute le modèle à ce compartiment. Si vous n'avez pas encore CloudFormation créé de compartiment, il crée un compartiment unique pour chaque région dans laquelle vous téléchargez un fichier modèle.

Les points suivants doivent être pris en compte lors de l'utilisation des compartiments S3 créés par CloudFormation :

- Les compartiments sont accessibles à toute personne disposant d'autorisations Amazon S3 dans votre Compte AWS compte.
- CloudFormation crée les compartiments avec le chiffrement côté serveur activé par défaut, chiffrant ainsi tous les objets stockés dans le compartiment.

Vous pouvez gérer directement les options de chiffrement pour les compartiments créés, par exemple, à l'aide de la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>, ou du AWS CLI. CloudFormation Pour plus d'informations, consultez [Configuration du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

- Pour utiliser votre propre compartiment et gérer ses autorisations, vous pouvez importer manuellement les modèles dans Amazon S3. Lorsque vous créez ou mettez à jour une pile, spécifiez l'URL Amazon S3 d'un fichier de modèle.
- Si vous n'avez pas encore de modèle prêt, vous pouvez choisir Créer avec Infrastructure Composer pour créer un modèle à l'aide d'Infrastructure Composer. Pour de plus amples informations, veuillez consulter [Compositeur d'infrastructure](#).

5. Choisissez Suivant pour continuer et valider le modèle.

Avant de continuer, CloudFormation valide votre modèle pour détecter les erreurs syntaxiques et certaines erreurs sémantiques, telles que les dépendances circulaires. Lors de la validation, CloudFormation vérifie d'abord si le modèle est un JSON valide. Si ce n'est pas le cas, CloudFormation vérifie si le modèle est YAML valide. Si les deux vérifications échouent, CloudFormation renvoie une erreur de validation du modèle.

6. Sur la page Specify stack details (Spécifier les détails de la pile), entrez un nom pour la pile dans la zone Nom de la pile.

Ce nom est un identifiant qui vous aidera à retrouver une pile particulière dans une liste de piles. Un nom de pile ne peut contenir que des caractères alphanumériques (sensibles à la casse) et des traits d'union. Il doit commencer par un caractère alphabétique et ne doit pas dépasser 128 caractères.

7. Dans la section Paramètres, les valeurs des paramètres définis dans le modèle.

8. Choisissez Suivant pour continuer la création de la pile.

9. (Facultatif) Sur la page Configurer les options de la pile, modifiez les options par défaut de la pile si nécessaire. Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#).

10. Si votre modèle contient des ressources IAM, dans la section Capacités, choisissez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous autorisez l'utilisation de ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).

11. Choisissez Next (Suivant) pour continuer.

12. Sur la page Vérifier et créer, vérifiez les détails de votre pile.

Pour modifier une valeur avant de lancer la pile, choisissez Modifier dans la section contenant le paramètre à modifier.

13. (Facultatif) Vous pouvez créer un jeu de modifications pour prévisualiser la configuration de la pile avant sa création. Sur la page Vérifier et créer, choisissez Créer un jeu de modifications et suivez les instructions. Pour de plus amples informations, veuillez consulter [Prévisualisation de la configuration de la pile](#).
14. Choisissez Soumettre pour lancer votre pile.

CloudFormation procédera ensuite à la création de toutes les ressources définies dans le modèle.

Vous pouvez suivre l'avancement et l'état de la création dans l'onglet Événements de votre nouvelle pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Création d'une pile à l'aide de la ligne de commande

Vous pouvez utiliser l'une des commandes suivantes :

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

Pour des exemples de création de pile à l'aide de la ligne de commande, consultez [Exemples de commandes d'opération de CloudFormation pile pour AWS CLI et PowerShell](#).

Configuration des options de pile

Sur la page Configurer les options de pile, vous pouvez configurer des options pour vos CloudFormation piles, telles que les balises, la notification des événements de pile ou une politique de pile.

Vous pouvez définir les options de pile suivantes :

Balises

Vous pouvez ajouter jusqu'à 50 paires de clés de balises à votre pile et à toutes les ressources compatibles CloudFormation avec le balisage pour. Une balise est une clé et une valeur définies par le client qui peuvent être attribuées aux AWS ressources à des fins telles que le suivi des coûts.

Une clé se compose de caractères alphanumériques et d'espaces. Les clés de balise peuvent contenir jusqu'à 127 caractères.

Une valeur se compose de caractères alphanumériques et d'espaces. Les valeurs de balise peuvent contenir jusqu'à 255 caractères.

Après la création de la pile, l'ajout, la mise à jour ou la suppression de balises au niveau de la pile déclenchera une mise à jour de la pile. Toutes les ressources prenant en charge la propagation des balises au niveau de la pile seront mises à jour en conséquence.

Autorisations

Rôle de service IAM existant qui CloudFormation peut assumer. Au lieu d'utiliser les informations d'identification de votre CloudFormation compte, utilisez les informations d'identification du rôle pour créer votre pile. Pour de plus amples informations, veuillez consulter [CloudFormation rôle de service](#).

Options d'échec de pile

Spécifie les options d'échec pour tous les déploiements de pile et les opérations d'ensembles de modifications. Pour de plus amples informations, veuillez consulter [Choisissez comment gérer les défaillances lors de l'approvisionnement des ressources](#).

L'option Roll back all stack resources (Annuler toutes les ressources de pile) permet de restaurer toutes les ressources spécifiées dans le modèle lorsque l'état de la pile est CREATE_FAILED ou UPDATE_FAILED.

Pour les opérations de création, l'option Preserve successfully provisioned resources (Préserver les ressources correctement allouées) préserve l'état des ressources réussies, tandis que les ressources en échec resteront à un état d'échec jusqu'à ce que la prochaine opération de mise à jour soit effectuée.

Pour les opérations de mise à jour et d'ensemble de modifications, l'option Preserve successfully provisioned resources (Préserver les ressources correctement allouées) permet de conserver l'état des ressources réussies tout en rétablissant les ressources défaillantes au dernier état stable connu. Les ressources en échec se trouveront à l'état UPDATE_FAILED. Les ressources sans dernier état stable connu seront supprimées lors de la prochaine opération de pile.

Vous pouvez également définir les options avancées suivantes pour la création de la pile :

Politique de pile

Définit les ressources que vous voulez protéger contre les mises à jour accidentelles pendant une mise à jour de pile. Par défaut, toutes les ressources peuvent être mises à jour pendant la mise à jour d'une pile.

Vous pouvez entrer la politique de pile directement au format JSON ou charger un fichier JSON contenant la politique de pile. Pour de plus amples informations, veuillez consulter [Empêcher les mises à jour des ressources de la pile](#).

Configuration de la restauration

Vous pouvez CloudFormation surveiller l'état de votre pile lors de sa création et de sa mise à jour, puis annuler cette opération si la pile dépasse le seuil de l'une des alarmes que vous avez spécifiées. Spécifiez les CloudWatch alarmes à CloudFormation surveiller. Si l'une des alarmes se déclenche pendant ALARM le fonctionnement de la pile ou pendant la période de surveillance, CloudFormation annule l'ensemble du fonctionnement de la pile. Pour de plus amples informations, veuillez consulter [Réduisez votre batterie en CloudFormation cas de violation d'alarme grâce à des déclencheurs](#).

Options de notification

Vous pouvez spécifier une rubrique Amazon Simple Notification Service nouvelle ou existante où seront envoyées les notifications sur les événements de pile.

Si vous créez une rubrique Amazon SNS, vous devez spécifier un nom et une adresse e-mail, où devront être envoyées les notifications d'événement de la pile.

Options de création de pile

Les options suivantes sont incluses pour la création d'une pile, mais ne sont pas disponibles dans le cadre des mises à jour des piles.

Expiration

Spécifie le temps, en minutes, qui CloudFormation doit être alloué avant la fin des opérations de création de pile. Si vous ne CloudFormation pouvez pas créer la pile complète dans le temps imparti, la création de la pile échoue en raison du délai imparti et annule la pile.

Par défaut, il n'y a aucun délai d'expiration pour la création de la pile. Cependant, les ressources individuelles peuvent avoir leur propre délai d'expiration en fonction de la nature du service qu'elles mettent en œuvre. Par exemple, si une ressource individuelle dans votre pile

arrive à expiration, la création de la pile arrive aussi à expiration, même si le délai que vous avez spécifié pour la création de la pile n'a pas encore été atteint.

Protection de la résiliation

Empêche la suppression accidentelle d'une pile. Si un utilisateur tente de supprimer une pile pour laquelle la protection contre la résiliation est activée, la suppression échoue et la pile, et son statut, restent inchangés. Pour de plus amples informations, veuillez consulter [CloudFormation Empêcher les piles d'être supprimées](#).

La protection de la résiliation est Désactivée par défaut.

Prévisualisation de la configuration de la pile

Pour prévisualiser la configuration d'une CloudFormation pile avant de la créer, créez un ensemble de modifications. Cette fonctionnalité vous permet d'examiner différentes configurations et d'apporter des corrections et des modifications à votre pile avant d'exécuter le jeu de modifications. Pour de plus amples informations sur les jeux de modifications, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

Création d'un jeu de modifications pour une nouvelle pile

Pour créer un jeu de modifications pour une nouvelle pile, sélectionnez votre modèle de pile et spécifiez la configuration de votre pile comme vous le feriez si vous créez une nouvelle pile, puis choisissez de créer un nouveau jeu de modifications au lieu d'une nouvelle pile.

Création d'un jeu de modifications pour une nouvelle pile

1. Sur la page Vérifier et créer, choisissez Créer.
2. Dans la boîte de dialogue Créer un jeu de modifications, saisissez un nom pour le jeu de modifications et, si vous le souhaitez, une description. Choisissez Create change set (Créer un jeu de modifications).

Lorsque vous créez un ensemble de modifications pour une nouvelle pile, CloudFormation procède comme suit :

- Il lance une nouvelle pile avec le statut REVIEW_IN_PROGRESS.
- Il crée un jeu de modifications pour la nouvelle pile qui reflète la configuration de pile que vous avez spécifiée dans les étapes précédentes.

CloudFormation affiche la page Change sets pour la pile proposée. Lors de la création de l'ensemble de modifications, son statut est `CREATE_IN_PROGRESS` et son état d'exécution est `UNAVAILABLE`. Une fois la création de l'ensemble de modifications CloudFormation terminée, il définit le statut de l'ensemble de modifications sur `CREATE_COMPLETE`, et son statut d'exécution est défini sur `AVAILABLE`. Le statut de la pile est mis à jour sur `REVIEW_IN_PROGRESS`. À ce stade, vous pouvez exécuter le jeu de modifications pour terminer la création de la nouvelle pile.

Dans le volet Changes, CloudFormation affiche la configuration proposée pour votre stack.

Si la création de l'ensemble de modifications CloudFormation échoue, le statut du jeu de modifications est défini sur `CREATE_FAILED`. Corrigez l'erreur affichée dans le champ Motif du statut, puis créez un nouveau jeu de modifications. À ce stade, vous pouvez essayer différentes configurations et apporter des corrections et des modifications à votre pile avant d'exécuter le jeu de modifications suivant.

3. Pour terminer la création d'une pile en fonction du jeu de modifications, choisissez **Execute** (Exécuter), spécifiez votre configuration de restauration, puis choisissez **Execute change set** (Exécuter un jeu de modifications).

Afficher les informations relatives à la pile depuis la CloudFormation console

Après avoir créé une CloudFormation pile, vous pouvez l'utiliser AWS Management Console pour afficher ses données et ses ressources.

Pour consulter les informations relatives à votre CloudFormation stack

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de la pile.
3. Sur la page Stacks, choisissez le nom de la pile. CloudFormation affiche les détails de la pile sélectionnée.

 Note

Si la pile a été supprimée, vous pouvez la retrouver en utilisant l'option Filtrer par état. Pour de plus amples informations, veuillez consulter [Afficher les piles supprimées depuis la console CloudFormation](#).

Les informations de pile suivantes sont disponibles :

Stack info (Infos de pile)

Affiche des informations générales sur la pile et sa configuration, y compris :

Présentation de

Affiche le nom de la pile, l'ID de la pile, la pile racine et le [rôle IAM](#), ainsi que des informations d'état telles que l'état de la pile, l'état de dérive et la protection contre la résiliation.

Étiquettes

Affiche toutes les balises associées à la pile.

Politique de pile

Décrit les ressources qui sont protégées contre les mises à jour de la pile. Pour mettre à jour ces ressources, elles doivent être explicitement autorisées pendant une mise à jour de la pile. Pour de plus amples informations, veuillez consulter [Empêcher les mises à jour des ressources de la pile](#).

Configuration de la restauration

Affiche toutes les CloudWatch alarmes que vous avez spécifiées et qui CloudFormation doivent être surveillées pendant le fonctionnement de la pile ou pendant la période de surveillance spécifiée. Si l'une des alarmes se déclenche pendant ALARM le fonctionnement de la pile ou pendant la période de surveillance, CloudFormation annule l'ensemble du fonctionnement de la pile. Pour de plus amples informations, veuillez consulter [Réduisez votre batterie en CloudFormation cas de violation d'alarme grâce à des déclencheurs](#).

Options de notification

Affiche la rubrique Amazon Simple Notification Service dans laquelle sont envoyées les notifications d'événement, si elle est spécifiée.

Événements

Affiche les opérations qui sont suivies lorsque vous créez, mettez à jour ou supprimez la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Tous les événements déclenchés par une opération de pile donnée reçoivent le même jeton de demande de client, lequel peut être utilisé pour suivre les opérations. Les opérations de pile lancées depuis la console utilisent le format de jeton Console- StackOperation -ID, qui vous permet d'identifier facilement l'opération de pile. Par exemple, si vous créez une pile à l'aide de la console, chaque événement de pile résultant reçoit le même jeton au format suivant : Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Ressources

Affiche les ressources qui font partie de la pile.

Sorties

Affiche les sorties qui ont été déclarées dans le modèle de la pile. Pour de plus amples informations, veuillez consulter [Obtenir des sorties exportées à partir d'une CloudFormation pile déployée](#).

Parameters

Affiche les paramètres de la pile et leurs valeurs.

Pour les piles contenant des paramètres Systems Manager, la colonne Valeur résolue affiche les valeurs utilisées dans la définition de la pile pour ces paramètres Systems Manager. Pour de plus amples informations, veuillez consulter [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#).

Modèle

Affiche le modèle de la pile.

Pour les piles contenant des macros, choisissez Afficher le modèle original pour afficher le modèle soumis par l'utilisateur, ou Afficher le modèle traité pour afficher le modèle après avoir traité CloudFormation les macros référencées. CloudFormation utilise le modèle traité pour créer ou mettre à jour votre pile.

Jeux de modifications

Affiche les jeux de modifications de la pile.

Pour de plus amples informations, veuillez consulter [Afficher un ensemble de modifications pour une CloudFormation pile](#).

Synchronisation depuis Git

Affiche le tableau de bord de synchronisation depuis Git de la pile.

Pour de plus amples informations, veuillez consulter [Tableau de bord d'état de la synchronisation Git](#).

Mettre à jour votre modèle de pile

Pour modifier les ressources ou les propriétés d'une CloudFormation pile, vous devez mettre à jour le modèle de la pile. Commencez par le modèle existant pour cette pile et apportez-y vos modifications. Si vous avez stocké le modèle dans un système de contrôle de source, utilisez une copie de celui-ci comme point de départ. Sinon, vous pouvez obtenir une copie du modèle auprès de CloudFormation.

Si vous voulez uniquement modifier les paramètres ou les réglages de la pile (comme la rubrique Amazon SNS d'une pile), vous pouvez réutiliser le modèle existant sans en obtenir de copie.

Vous pouvez mettre à jour un modèle de CloudFormation pile à l'aide d'un éditeur de texte ou [Compositeur d'infrastructure](#).

Pour mettre à jour un modèle de pile existant à l'aide d'Infrastructure Composer

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Piles, sélectionnez le nom de la pile à mettre à jour.
3. Sélectionnez l'onglet Modèle, puis Afficher dans Infrastructure Composer.

CloudFormation ouvre le modèle dans Infrastructure Composer.

4. Mettez à jour votre modèle à l'aide de l'une des méthodes suivantes :
 - Interface Canevas : ici, vous pouvez glisser-déposer à partir de la palette Ressources. Configurez les ressources en double-cliquant sur une carte pour ouvrir le panneau Propriétés de la ressource. Connectez les ressources selon vos besoins. Pour obtenir des instructions détaillées sur l'utilisation de l'interface Canevas, consultez [Comment composer dans AWS Infrastructure Composer](#).

- Interface Modèle : passez de l'interface Canevas à l'interface Modèle. Effectuez des mises à jour en ligne du code du modèle. Basculez entre les formats JSON et YAML selon vos besoins.
5. Sélectionnez Valider pour vérifier qu'il n'y a pas d'erreurs de syntaxe dans le modèle.
 6. Lorsque vous êtes prêt à exporter les modifications vers CloudFormation, choisissez Mettre à jour le modèle.

Pour mettre à jour un modèle de pile existant à l'aide du AWS CLI

1. Pour obtenir le modèle de la pile que vous voulez mettre à jour, utilisez la commande CLI [get-template](#).
2. Copiez le modèle, collez-le dans un fichier texte, modifiez-le et enregistrez-le. Copiez uniquement le modèle. La commande délimite le modèle par des guillemets. Ne copiez pas ces guillemets. Le modèle lui-même commence par une accolade d'ouverture et se termine par une accolade de fermeture. Spécifiez les modifications que vous souhaitez apporter aux ressources de la pile dans ce fichier.

Veillez garder à l'esprit les points suivants lorsque vous apportez des modifications à votre modèle :

- Vous ne pouvez pas ajouter, modifier ou supprimer un paramètre utilisé par une ressource qui ne prend pas en charge les mises à jour.
- Pour la plupart des ressources, la modification de leur nom logique revient à supprimer les ressources correspondantes et à les remplacer par une nouvelle. Toutes les autres ressources qui dépendent de la ressource renommée doivent également être mises à jour et peuvent entraîner leur remplacement. D'autres ressources nécessitent que vous mettiez à jour une propriété (non seulement le nom logique) pour lancer une mise à jour.
- Certaines ressources peuvent avoir des contraintes quant aux valeurs que vous pouvez définir pour certaines propriétés. Par exemple, les modifications apportées à la propriété `AllocatedStorage` d'une instance de base de données RDS doivent être supérieures à la valeur actuelle. Si votre mise à jour enfreint ces règles, cette partie échouera.
- La mise à jour d'une ressource peut également affecter d'autres ressources qui la référencent. Si vous utilisez des fonctions telles que [La fonction Ref](#) ou [La fonction Fn::GetAtt](#) pour définir une propriété basée sur une autre ressource, la ressource de référence CloudFormation sera également mise à jour lorsque la ressource référencée changera.

- Pour plus d'informations sur les effets de la mise à jour de propriétés de ressources particulières, consultez la [Référence des types de ressources et de propriétés AWS](#). Pour chaque propriété, une mise à jour peut avoir l'un des impacts suivants :
 - Requis pour la mise à jour : [aucune interruption](#)
 - Requis pour la mise à jour : [quelques interruptions](#)
 - Requis pour la mise à jour : un [remplacement](#)
- Vous pouvez vérifier la syntaxe JSON ou YAML de votre modèle à l'aide de la commande CLI [validate-template](#) ou en spécifiant votre modèle sur la console. La console effectue automatiquement la validation. Cependant, ces méthodes vérifient uniquement la syntaxe de votre modèle et ne valident pas la validité des valeurs de propriété que vous avez spécifiées pour une ressource. Pour des validations plus complexes ou pour vérifier les meilleures pratiques, vous pouvez également utiliser des outils supplémentaires tels que [CloudFormation Linter \(cfn-lint\)](#) et [CloudFormation Rain \(rain fmt\)](#).

Note

Parfois, il CloudFormation n'autorise pas certaines modifications que vous essayez d'apporter et cela vous indique que le changement n'est pas autorisé. Ce message peut toutefois s'afficher de manière asynchrone, car les ressources sont créées et mises à jour par défaut CloudFormation dans un ordre non déterministe.

Comprendre les comportements de mise à jour des ressources d'une pile

Lorsque vous soumettez une mise à jour, les ressources sont CloudFormation mises à jour en fonction des différences entre ce que vous soumettez et le modèle actuel de la pile. Les ressources qui n'ont pas changé continuent à être exécutées sans interruption pendant le processus de mise à jour. Pour les ressources mises à jour, CloudFormation utilise l'un des comportements de mise à jour suivants :

Mise à jour sans interruption

CloudFormation met à jour la ressource sans perturber le fonctionnement de cette ressource et sans modifier l'identifiant physique de la ressource. Par exemple, si vous mettez à jour certaines

propriétés d'une [AWS::CloudTrail::Trail](#) ressource, le CloudFormation suivi est mis à jour sans interruption.

Mises à jour avec quelques interruptions

CloudFormation met à jour la ressource avec quelques interruptions. Par exemple, si vous mettez à jour certaines propriétés d'une [AWS::EC2::Instance](#) ressource, l'instance peut être interrompue pendant qu' Amazon EC2 la reconfigure.

Remplacement

CloudFormation recrée la ressource lors d'une mise à jour, ce qui génère également un nouvel identifiant physique. CloudFormation crée généralement d'abord la ressource de remplacement, modifie les références des autres ressources dépendantes pour qu'elles pointent vers la ressource de remplacement, puis supprime l'ancienne ressource. Par exemple, si vous mettez à jour la AvailabilityZone propriété d'un type de [AWS::EC2::Instance](#) ressource, CloudFormation crée une nouvelle ressource et remplace la ressource d'instance EC2 actuelle par la nouvelle.

Si vous ajoutez ou supprimez une propriété qui nécessite un remplacement, cela déclenchera également une mise à jour. La mise à jour se produira même si la valeur réelle de la propriété ne change pas.

La méthode CloudFormation utilisée dépend de la propriété que vous mettez à jour pour un type de ressource donné. Le comportement de mise à jour de chaque propriété est décrit dans la [Référence des types de ressources et de propriétés AWS](#).

Selon le comportement de mise à jour, vous pouvez décider quand modifier les ressources afin de limiter l'impact de ces modifications sur votre application. En particulier, vous pouvez planifier le moment où les ressources doivent être remplacées au cours d'une mise à jour. Par exemple, si vous mettez à jour la Port propriété d'un type de DBInstance ressource [AWS::RDS::DBInstance](#), remplacez l'instance de base de CloudFormation données en créant une nouvelle instance de base de données avec le paramètre de port mis à jour et supprimez l'ancienne instance de base de données. Avant la mise à jour, envisagez d'effectuer les opérations suivantes pour vous préparer au remplacement de la base de données :

- Créez un instantané des bases de données actuelles.
- Préparez une stratégie définissant la façon dont les applications qui utilisent cette instance DB doivent gérer une interruption lorsque l'instance est remplacée.

- Assurez-vous que les applications qui utilisent cette instance DB prennent en compte le paramètre de port mis à jour et les autres mises à jour que vous avez effectuées.
- Utilisez l'instantané DB pour restaurer les bases de données dans la nouvelle instance de base de données.

Cet exemple n'est pas exhaustif. Il a pour objectif de vous donner une idée des éléments à prévoir lorsqu'une ressource est remplacée pendant une mise à jour.

Note

Si le modèle inclut une ou plusieurs [piles imbriquées](#), lance CloudFormation également une mise à jour pour chaque pile imbriquée. Cela est nécessaire pour déterminer si les piles imbriquées ont été modifiées. CloudFormation met à jour uniquement les ressources des piles imbriquées dont les modifications sont spécifiées dans les modèles correspondants.

Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications

Avant de mettre à jour une pile, il peut être utile de comprendre l'incidence des modifications sur les ressources exécutées. Les jeux de modifications vous permettent de prévisualiser l'impact des modifications proposées sur une pile sur vos ressources en cours d'exécution, y compris l'impact sur les propriétés et les attributs des ressources. Si vos modifications supprimeront ou remplaceront des ressources critiques, CloudFormation apportez les modifications à votre pile uniquement lorsque vous décidez d'exécuter l'ensemble de modifications, ce qui vous permet de décider de poursuivre les modifications proposées ou d'explorer d'autres modifications en créant un autre ensemble de modifications. Vous pouvez créer et gérer des ensembles de modifications à l'aide de la CloudFormation console ou de CloudFormation l'API. AWS CLI

Rubriques

- [Création d'un ensemble de modifications pour une CloudFormation pile](#)
- [Afficher un ensemble de modifications pour une CloudFormation pile](#)
- [Utilisation d'ensembles de modifications sensibles à la dérive](#)
- [Exécuter un ensemble de modifications pour une CloudFormation pile](#)
- [Supprimer un ensemble de modifications pour une CloudFormation pile](#)

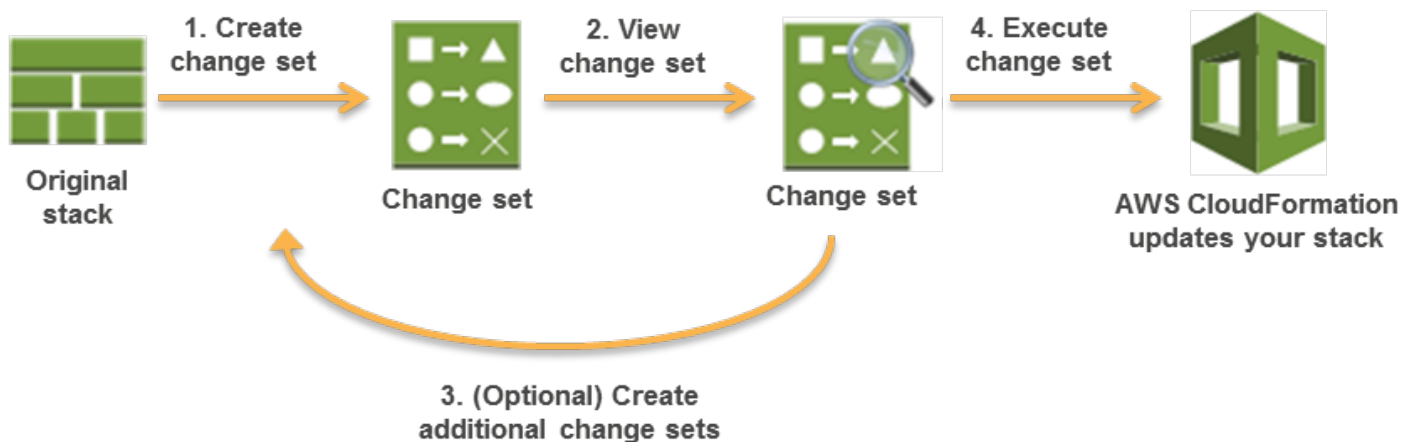
- [Exemples d'ensembles de modifications pour les CloudFormation piles](#)
- [Modifier les ensembles pour les piles imbriquées](#)

⚠ Important

Les ensembles de modifications n'indiquent pas si la mise à jour d'une pile CloudFormation sera réussie. Par exemple, un jeu de modifications ne vérifie pas si vous allez dépasser le quota de votre compte, si vous mettez à jour une ressource qui ne prend pas en charge les mises à jour ou si vous disposez des autorisations suffisantes pour modifier une ressource, autant d'éléments qui peuvent entraîner l'échec de la mise à jour d'une pile. En cas d'échec de mise à jour, CloudFormation tente de restaurer l'état d'origine de vos ressources.

Présentation des jeux de modifications

Le schéma suivant résume comment utiliser les jeux de modifications pour mettre à jour une pile :



1. Pour créer un jeu de modifications, vous devez soumettre les modifications que vous souhaitez apporter à la pile. Vous pouvez soumettre un modèle de pile modifié ou des valeurs de paramètres d'entrée modifiées. CloudFormation compare votre pile avec les modifications que vous avez soumises pour générer l'ensemble de modifications ; elle n'apporte aucune modification à votre pile pour le moment.
2. Affichez le jeu de modifications pour identifier les paramètres et les ressources de la pile qui seront modifiés. Par exemple, vous pouvez voir quelles ressources CloudFormation seront ajoutées, modifiées ou supprimées. En outre, vous pouvez voir une before-and-after comparaison des propriétés et des attributs des ressources, tels que les balises, qui CloudFormation seront modifiés.

3. Facultatif : Si vous souhaitez prendre en compte d'autres modifications avant de choisir les changements à apporter, créez des jeux de modifications supplémentaires. La création de plusieurs jeux de modifications vous aide à comprendre et à évaluer l'impact des différentes modifications sur vos ressources et vos propriétés. Vous pouvez créer autant de jeux de modifications que vous le souhaitez.
4. Exécutez le jeu de modifications qui contient les modifications que vous souhaitez appliquer à votre pile. CloudFormation met à jour votre pile avec ces modifications.

 Note

Après avoir exécuté une modification, CloudFormation supprime tous les ensembles de modifications associés à la pile car ils ne sont pas applicables à la pile mise à jour.

Vous pouvez également supprimer des jeux de modifications si vous ne souhaitez pas qu'ils soient appliqués.

Création d'un ensemble de modifications pour une CloudFormation pile

Pour créer un ensemble de modifications pour une pile en cours d'exécution, soumettez les modifications que vous souhaitez apporter en fournissant un modèle modifié, de nouvelles valeurs de paramètres d'entrée, ou les deux. CloudFormation génère un ensemble de modifications en comparant votre pile aux modifications que vous avez soumises.

Vous pouvez modifier un modèle avant de créer le jeu de modifications ou lors de la création du jeu de modifications.

Create a change set (console)

Pour créer un jeu de modifications

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez la pile en cours d'exécution que vous voulez créer un jeu de modifications.
4. Dans le volet des détails de la pile, sélectionnez Actions de la pile, puis Créer un jeu de modifications.

5. Sur la *stack-name* page Créer un ensemble de modifications pour, effectuez l'une des opérations suivantes pour modifier les valeurs des paramètres d'entrée, spécifier l'emplacement d'un modèle mis à jour ou modifier le modèle :

Sous-tâche	Action
Pour modifier les valeurs des paramètres d'entrée	Sélectionnez Utiliser le modèle existant, puis Suivant pour continuer à saisir ou modifier les valeurs des paramètres d'entrée.
Pour spécifier l'emplacement d'un modèle mis à jour	Si vous avez modifié le modèle, sélectionnez Remplacer le modèle existant, puis effectuez l'une des opérations suivantes : - Pour un modèle stocké dans un compartiment Amazon S3, choisissez URL Amazon S3. Entrez ou collez l'URL du modèle, puis choisissez Suivant. Si votre modèle se trouve dans un compartiment dans lequel la gestion des versions est activée, vous pouvez spécifier une version particulière du modèle en ajoutant ?versionId= <i>version-id</i> à l'URL. Pour plus d'informations, consultez Travail avec des objets dans un compartiment avec gestion des versions activée dans le Guide de l'utilisateur Amazon Simple Storage Service. - Pour un modèle stocké localement sur votre ordinateur, choisissez Upload a template file (Charger un fichier de modèle). Choisissez Choose File (Choisir un fichier) pour accéder au fichier et le sélectionner, puis choisissez Suivant.
Pour modifier le modèle	Si vous n'avez pas modifié le modèle, sélectionnez Modifier le modèle dans Infrastructure Composer, puis sélectionnez Modifier dans Infrastructure Composer. Vous êtes redirigé vers AWS Infrastructure Composer. Une fois que vous avez modifié le modèle, choisissez Créer un ensemble de modifications, puis Confirmer et CloudFormation continuer pour revenir à la <i>stack-name</i> page Créer un ensemble de modifications pour, puis choisissez Suivant.

6. Sur la page Spécifier les détails de la pile, spécifiez un nom pour le jeu de modifications et, si vous le souhaitez, spécifiez une description du jeu de modifications afin d'identifier son objectif dans la section Vue d'ensemble. Si votre modèle contient des paramètres, sur la page Specify stack details (Spécifier les détails de la pile), entrez ou modifiez les valeurs des paramètres d'entrée applicables, puis choisissez Suivant.

Si vous réutilisez le modèle de la pile, CloudFormation renseigne chaque paramètre avec la valeur actuelle de la pile, à l'exception des paramètres déclarés avec l'NoEchoattribut. Pour utiliser les valeurs existantes pour ces paramètres, sélectionnez Utiliser valeur existante.

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles et sur l'utilisation de paramètres dynamiques pour gérer les secrets, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

7. Sur la page Configurer les options de la pile, mettez à jour les balises de la pile, le rôle de service IAM, la politique de la pile, la configuration de restauration, le sujet de notification Amazon SNS (le cas échéant) ou les jeux de modifications.

 Note

Les ensembles de modifications pour les piles imbriquées sont Activés par défaut, ce qui crée des ensembles de modifications pour toutes les piles imbriquées spécifiées dans votre modèle. Pour créer un ensemble de modifications pour la pile active uniquement, choisissez Désactivé. Pour plus d'informations sur les jeux de modifications pour les piles imbriquées, consultez [Modifier les ensembles pour les piles imbriquées](#).

8. Si le modèle inclut des ressources IAM, dans Fonctionnalités, sélectionnez Je reconnais que cela CloudFormation pourrait créer des ressources IAM. Les ressources IAM peuvent modifier les autorisations dans votre compte AWS . Consultez ces ressources pour vous assurer que vous autorisez uniquement les actions que vous prévoyez. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
9. Choisissez Next (Suivant) pour continuer.
10. Sur la *stack-name* page Révision, passez en revue les modifications apportées à cet ensemble de modifications.
11. Sélectionnez Soumettre.

Vous êtes redirigé vers l'onglet Modifications de la page des détails du jeu de modifications. Lors de CloudFormation la génération de l'ensemble de modifications, le statut de l'ensemble de modifications est `CREATE_PENDING`. Après avoir créé l'ensemble de modifications, CloudFormation définit le statut sur `CREATE_COMPLETE`. Dans la section Modifications, CloudFormation répertorie toutes les modifications qu'il apportera à votre pile. Pour de plus amples informations, veuillez consulter [Afficher un ensemble de modifications pour une CloudFormation pile](#).

Sélectionnez Afficher les détails dans la colonne Modifications au niveau des propriétés pour afficher les modifications apportées au niveau des propriétés.

Si vous CloudFormation ne parvenez pas à créer l'ensemble de modifications (`FAILED` État des rapports), corrigez l'erreur affichée dans le champ État, puis recréez le jeu de modifications.

- Après avoir vérifié que les modifications sont correctes, sélectionnez Exécuter le jeu de modifications

Create a change set for nested stacks (console)

Pour créer un jeu de modifications pour des piles imbriquées

- Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
- Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
- Sur la page Piles, sélectionnez la pile en cours d'exécution pour laquelle vous voulez créer un jeu de modifications.
- Dans le volet des détails de la pile, sélectionnez Actions de la pile, puis Créer un jeu de modifications.
- Sur la *stack-name* page Créer un ensemble de modifications pour, effectuez l'une des opérations suivantes pour modifier les valeurs des paramètres d'entrée, spécifier l'emplacement d'un modèle mis à jour ou modifier le modèle :

Sous-tâche	Action
Pour modifier les valeurs des paramètres d'entrée	Sélectionnez Utiliser le modèle existant, puis Suivant pour continuer à saisir ou modifier les valeurs des paramètres d'entrée.
Pour spécifier l'emplacement d'un modèle mis à jour	Si vous avez modifié le modèle, sélectionnez Remplacer le modèle existant, puis effectuez l'une des opérations suivantes : - Pour un modèle stocké dans un compartiment Amazon S3, choisissez URL Amazon S3. Entrez ou collez l'URL du modèle, puis choisissez Suivant. Si votre modèle se trouve dans un compartiment dans lequel la gestion des versions est activée, vous pouvez spécifier une version particulière du modèle en ajoutant ? versionId= <i>version-id</i> à l'URL. Pour plus d'informations, consultez Travail avec des objets dans un compartiment avec gestion des versions activée dans le Guide de l'utilisateur Amazon Simple Storage Service. - Pour un modèle stocké localement sur votre ordinateur, choisissez Upload a template file (Charger un fichier de modèle). Choisissez Choose File (Choisir un fichier) pour accéder au fichier et le sélectionner, puis choisissez Suivant.
Pour modifier le modèle	Si vous n'avez pas modifié le modèle, sélectionnez Modifier le modèle dans Infrastructure Composer, puis sélectionnez Modifier dans Infrastructure Composer. Vous êtes redirigé vers AWS Infrastructure Composer. Une fois que vous avez modifié le modèle, choisissez Créer un ensemble de modifications, puis Confirmer et CloudFormation continue pour revenir à la <i>stack-name</i> page Créer un ensemble de modifications pour, puis choisissez Suivant.

- Sur la page Spécifier les détails de la pile, spécifiez un nom pour le jeu de modifications et, si vous le souhaitez, spécifiez une description du jeu de modifications afin d'identifier son objectif dans la section Vue d'ensemble. Si votre modèle contient des paramètres, sur la

page Specify stack details (Spécifier les détails de la pile), entrez ou modifiez les valeurs des paramètres d'entrée applicables, puis choisissez Suivant.

Si vous réutilisez le modèle de la pile, CloudFormation renseigne chaque paramètre avec la valeur actuelle de la pile, à l'exception des paramètres déclarés avec l'NoEchoattribut. Pour utiliser les valeurs existantes pour ces paramètres, sélectionnez Utiliser valeur existante.

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles, ainsi que sur l'utilisation de paramètres dynamiques pour gérer les secrets, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

7. Sur la page Configurer les options de la pile, mettez à jour les balises de la pile, le rôle de service IAM, la politique de la pile, la configuration de restauration, le sujet de notification Amazon SNS (le cas échéant) ou les jeux de modifications. Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#).

 Note

Les ensembles de modifications pour les piles imbriquées sont Activés par défaut, ce qui crée des ensembles de modifications pour toutes les piles imbriquées spécifiées dans votre modèle. Pour plus d'informations sur les jeux de modifications pour les piles imbriquées, consultez [Modifier les ensembles pour les piles imbriquées](#).

8. Si le modèle inclut des ressources IAM, dans Fonctionnalités, sélectionnez Je reconnais que cela CloudFormation pourrait créer des ressources IAM. Les ressources IAM peuvent modifier les autorisations dans votre compte AWS . Consultez ces ressources pour vous assurer que vous autorisez uniquement les actions que vous prévoyez. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
9. Choisissez Next (Suivant) pour continuer.
10. Sur la *stack-name* page Révision, passez en revue les modifications apportées à cet ensemble de modifications.
11. Sélectionnez Soumettre.

 Note

CloudFormation les ensembles de modifications au niveau de la propriété ne résolvent pas les références entre piles lorsque vous créez des ensembles de

modifications pour des piles imbriquées. Les jeux de modifications peuvent marquer les ressources d'une pile enfant pour un remplacement conditionnel si elles font référence à la sortie d'une pile parent et que la pile parent a été modifiée.

Vous êtes redirigé vers l'onglet Modifications de la page des détails du jeu de modifications. Lors de CloudFormation la génération de l'ensemble de modifications, le statut de l'ensemble de modifications est `CREATE_PENDING`. Après avoir créé l'ensemble de modifications, CloudFormation définit le statut sur `CREATE_COMPLETE`. Dans la section Modifications, CloudFormation répertorie toutes les modifications qu'il apportera à votre pile. Pour de plus amples informations, veuillez consulter [Afficher un ensemble de modifications pour une CloudFormation pile](#).

Si vous CloudFormation ne parvenez pas à créer l'ensemble de modifications (FAILED état des rapports), corrigez l'erreur affichée dans le champ État, puis recréez le jeu de modifications.

12. Après avoir vérifié que les modifications sont correctes, sélectionnez Exécuter le jeu de modifications

Pour créer un ensemble de modifications (AWS CLI)

- Utilisez la commande [create-change-set](#).

Vous soumettez vos modifications en tant qu'options de commande. Vous pouvez spécifier de nouvelles valeurs de paramètre, un modèle mis à jour ou les deux. Par exemple, la commande suivante crée un jeu de modifications nommé `SampleChangeSet` pour la pile `MyStack`. Ce jeu de modifications utilise le modèle de la pile actuelle, mais avec une autre valeur pour le paramètre `Purpose` :

```
aws cloudformation create-change-set --stack-name MyStack \  
  --change-set-name SampleChangeSet --use-previous-template \  
  --parameters \  
    ParameterKey="InstanceType",UsePreviousValue=true  
    ParameterKey="KeyPairName",UsePreviousValue=true  
    ParameterKey="Purpose",ParameterValue="production"
```

Afficher un ensemble de modifications pour une CloudFormation pile

Après avoir créé un jeu de modifications, vous pouvez afficher les modifications proposées avant de les exécuter. Vous pouvez utiliser la CloudFormation console ou AWS CLI l' CloudFormation API pour afficher les ensembles de modifications. La console CloudFormation propose un récapitulatif des modifications, ainsi qu'une liste détaillée des modifications au format JSON. L' AWS CloudFormation API AWS CLI and renvoie une liste détaillée des modifications au format JSON.

View a change set (console)

Pour afficher un jeu de modifications

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez le nom de la pile contenant le jeu de modifications que vous voulez afficher.
4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Choisissez le nom de l'ensemble de modifications que vous souhaitez afficher.

La CloudFormation console vous dirige vers la page de détails de l'ensemble de modifications, où vous pouvez voir l'heure à laquelle l'ensemble de modifications a été créé, son statut, les entrées utilisées pour générer l'ensemble de modifications et un résumé des modifications.

Dans la section Modifications , chaque ligne représente une ressource que CloudFormation va ajouter, modifier ou retirer.

- Ajouter : CloudFormation crée une ressource lorsque vous ajoutez une ressource au modèle de la pile.
- Modifier : CloudFormation modifie une ressource lorsque vous modifiez les propriétés d'une ressource dans le modèle de la pile.
- Supprimer : CloudFormation supprime une ressource lorsque vous supprimez une ressource du modèle de la pile.

 Note

Une modification peut entraîner l'interruption ou le remplacement (ou la recreation) de la ressource. Pour plus d'informations sur les mises à jour de ressources, consultez [Comprendre les comportements de mise à jour des ressources d'une pile](#).

Pour cibler des modifications spécifiques, utilisez la vue filtre. Par exemple, filtrez un type de ressource spécifique, tel que `AWS::EC2::Instance`. Pour filtrer une ressource spécifique, spécifiez son ID physique ou logique, tel que `myWebServer` ou `i-123abcd4`.

6. Dans la section Modifications, sélectionnez Afficher les détails dans la colonne Modifications au niveau des propriétés pour afficher les modifications apportées aux valeurs des propriétés de votre ressource.
7. La CloudFormation console vous dirige vers la page des modifications au niveau des propriétés d'une ressource, où vous pouvez voir la configuration du modèle de la ressource avant d'exécuter un ensemble de modifications et à quoi ressemblera la configuration du modèle après l'exécution de l'ensemble de modifications.

Le tableau de la section Modifications au niveau des propriétés affiche le Chemin d'accès, le Type de modification, la Valeur avant, et la Valeur après pour les propriétés concernées. Dans le tableau, cochez la case correspondant à chaque modification que vous voulez mettre en évidence dans les vues Avant et Après de votre modèle afin de voir les modifications qui seront apportées au niveau des propriétés.

- Ajouter : les propriétés ajoutées sont surlignées en vert.
- Modifier : les propriétés modifiées sont surlignées en bleu.
- Supprimer : les propriétés supprimées sont mises en évidence en rouge.

View a change set for nested stack (console)

Pour afficher un ensemble de modifications pour les piles imbriquées (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.

3. Sur la page Piles, sélectionnez le nom de la pile contenant le jeu de modifications que vous voulez afficher.
4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Choisissez le nom de l'ensemble de modifications que vous souhaitez afficher.

La CloudFormation console vous dirige vers la page de détails de l'ensemble de modifications, où vous pouvez voir l'heure à laquelle l'ensemble de modifications a été créé, son statut, les entrées utilisées pour générer l'ensemble de modifications et un résumé des modifications.

Dans la section Modifications, chaque ligne représente une ressource que CloudFormation va ajouter, modifier ou retirer ou dont il affichera l'état de dynamique.

- Ajouter : CloudFormation crée une ressource lorsque vous ajoutez une ressource au modèle de la pile.
- Modifier : CloudFormation modifie une ressource lorsque vous modifiez les propriétés d'une ressource dans le modèle de la pile.
- Supprimer : CloudFormation supprime une ressource lorsque vous supprimez une ressource du modèle de la pile.
- Dynamique : CloudFormation impossible de déterminer l'action exacte de modification des ressources à partir du modèle de la pile imbriquée.

 Note

Une modification peut entraîner l'interruption ou le remplacement (ou la recreation) de la ressource. Pour plus d'informations sur les mises à jour de ressources, consultez [Comprendre les comportements de mise à jour des ressources d'une pile](#).

Pour cibler des modifications spécifiques, utilisez la vue filtre. Par exemple, filtrez un type de ressource spécifique, tel que **AWS::CloudFormation::Stack**. Pour filtrer une ressource spécifique, spécifiez son ID physique ou logique, tel que **DeadLetterQueue** ou **NestedStack**.

6. Dans la section Modifications, choisissez Afficher l'ensemble de modifications imbriqué de l'ensemble de modifications imbriqué que vous souhaitez afficher.

La CloudFormation console vous dirige vers la page de détails de l'ensemble de modifications imbriqué. Vous pouvez choisir Accéder à l'ensemble de modifications racine pour afficher l'ensemble de modifications racine ou choisissez Afficher l'ensemble de modifications parent pour afficher l'ensemble de modifications parent. Pour plus d'informations, voir, [Modifier les ensembles pour les piles imbriquées](#).

 Note

CloudFormation les ensembles de modifications au niveau de la propriété ne résolvent pas les références entre piles lorsque vous créez des ensembles de modifications pour des piles imbriquées. Les jeux de modifications peuvent marquer les ressources d'une pile enfant pour un remplacement conditionnel si elles font référence à la sortie d'une pile parent et que la pile parent a été modifiée.

Pour afficher un ensemble de modifications (AWS CLI)

1. Pour obtenir l'ID du jeu de modifications, exécutez la commande [change-sets](#).

Indiquez le nom de la pile qui contient le jeu de modifications que vous voulez afficher, comme illustré dans l'exemple suivant :

```
aws cloudformation list-change-sets --stack-name MyStack
```

CloudFormation renvoie une liste d'ensembles de modifications, similaire à ce qui suit :

```
{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
      "Status": "CREATE_COMPLETE",
      "ChangeSetName": "SampleChangeSet",
      "CreationTime": "2020-11-18T20:44:05.889Z",
      "StackName": "MyStack",
      "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
    },
    {
```

```

    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-conditional",
    "CreationTime": "2020-11-18T21:15:56.398Z",
    "StackName": "MyStack",
    "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-conditional/1a2345b6-0000-00a0-
a123-00abc0abc000"
  },
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
    "Status": "CREATE_COMPLETE",
    "ChangeSetName": "SampleChangeSet-replacement",
    "CreationTime": "2020-11-18T21:03:37.706Z",
    "StackName": "MyStack",
    "ChangeSetId": "arn:aws:cloudformation:us-
east-1:123456789012:changeSet/SampleChangeSet-replacement/1a2345b6-0000-00a0-
a123-00abc0abc000"
  }
]
}

```

2. Exécutez la commande [describe-change-set](#), en spécifiant l'ID du jeu de modifications que vous souhaitez afficher. Par exemple :

```

aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000

```

CloudFormation renvoie des informations sur l'ensemble de modifications spécifié.

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    }
  ],

```

```
{
  "ParameterValue": "ellioty-useast1",
  "ParameterKey": "KeyPairName"
},
{
  "ParameterValue": "t2.micro",
  "ParameterKey": "InstanceType"
}
],
"Changes": [
  {
    "ResourceChange": {
      "ResourceType": "AWS::EC2::Instance",
      "PhysicalResourceId": "i-1abc23d4",
      "Details": [
        {
          "ChangeSource": "DirectModification",
          "Evaluation": "Static",
          "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never"
          }
        }
      ],
      "Action": "Modify",
      "Scope": [
        "Tags"
      ],
      "LogicalResourceId": "MyEC2Instance",
      "Replacement": "False"
    },
    "Type": "Resource"
  }
],
"CreationTime": "2020-11-18T23:35:25.813Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/9edde307-960d-4e6e-ad66-b09ea2f20255"
}
```

Utilisez `--include-property-values` avec `describe-change-set` pour répertorier les modifications au niveau des propriétés.

La clé `Changes` répertorie les modifications apportées aux ressources. Si vous deviez exécuter cet ensemble de modifications, les balises de l'instance `i-1abc23d4` EC2 CloudFormation seraient mises à jour. Pour obtenir une description de chaque champ, consultez le type de données [Change](#) dans la Référence d'API AWS CloudFormation .

Pour obtenir des exemples supplémentaires de jeux de modifications, consultez la page [Exemples d'ensembles de modifications pour les CloudFormation piles](#).

Pour afficher les modifications au niveau des propriétés dans un jeu de modifications (AWS CLI)

- La commande suivante répertorie les modifications au niveau des propriétés liées à un jeu de modifications pour une ressource `AWS::EC2::NetworkInterface` qui supprimera la propriété `Ipv4Prefixes`, modifiera le `Description` pour la ressource et ajoutera un `Tag` :

```
aws cloudformation describe-change-set --include-property-values \  
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841
```

Voici un exemple de sortie.

```
"ChangeSetName": "ExampleChangeSet",  
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",  
  "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/  
ab664180-f686-11ee-9e29-12cd92393671",  
  "StackName": "ExampleStack",  
  "Description": null,  
  "Parameters": null,  
  "CreationTime": "2024-04-09T18:04:59.935000+00:00",  
  "ExecutionStatus": "AVAILABLE",  
  "Status": "CREATE_COMPLETE",  
  "StatusReason": null,  
  "NotificationARNs": [],  
  "RollbackConfiguration": {  
    "RollbackTriggers": []  
  },  
  "Capabilities": [],
```



```

    "Tags": null,
    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
  {
    "Changes": [
      {
        "Type": "Resource",
        "ResourceChange": {
          "Action": "Modify",
          "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
          "PhysicalResourceId": "eni-067fd35b649a05b71",
          "ResourceType": "AWS::EC2::NetworkInterface",
          "Replacement": "False",
          "Scope": [
            "Properties",
            "Tags"
          ],
          "Details": [
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Ipv4Prefixes",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Ipv4Prefixes",
                "BeforeValue": "[]",
                "AttributeChangeType": "Remove"
              },
              "Evaluation": "Static",
              "ChangeSource": "DirectModification"
            },
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Description",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Description",
                "BeforeValue": "",
                "AfterValue": "Description",
                "AttributeChangeType": "Modify"
              },
              "Evaluation": "Static",

```

```

        "ChangeSource": "DirectModification"
    },
    {
        "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never",
            "Path": "/Properties/Tags/0",
            "AfterValue": "{\"Key\":\"Test\",\"Value\":\"Test\"}",
            "AttributeChangeType": "Add"
        },
        "Evaluation": "Static",
        "ChangeSource": "DirectModification"
    }
],
"BeforeContext": "{\"Properties\":{\"Description\":\"\",
\"PrivateIpAddress\":\"172.31.76.2\",\"PrivateIpAddresses\":[{\"PrivateIpAddress
\":\"172.31.76.2\",\"Primary\":\"true\"}],\"SecondaryPrivateIpAddressCount\":\"0\",
\"Ipv6PrefixCount\":\"0\",\"Ipv4Prefixes\":[],\"Ipv4PrefixCount\":\"0\",\"GroupSet
\":[\"sg-05a45689b1059e82d\"],\"Ipv6Prefixes\":[],\"SubnetId\":\"subnet-455e8969\",
\"SourceDestCheck\":\"true\",\"InterfaceType\":\"interface\",\"Tags\":[]},
\"UpdateReplacePolicy\":\"Retain\",\"DeletionPolicy\":\"Retain\"}",
"AfterContext": "{\"Properties\":{\"Description\":\"Description\",
\"PrivateIpAddress\":\"172.31.76.2\",\"PrivateIpAddresses\":[{\"PrivateIpAddress
\":\"172.31.76.2\",\"Primary\":\"true\"}],\"SecondaryPrivateIpAddressCount
\":\"0\",\"Ipv6PrefixCount\":\"0\",\"Ipv4PrefixCount\":\"0\",\"GroupSet\":
[\"sg-05a45689b1059e82d\"],\"Ipv6Prefixes\":[],\"SubnetId\":\"subnet-455e8969\",
\"SourceDestCheck\":\"true\",\"InterfaceType\":\"interface\",\"Tags\":[{\"Value\":
\"Test\",\"Key\":\"Test\"}]},\"UpdateReplacePolicy\":\"Retain\",\"DeletionPolicy\":
\"Retain\"}"
    }
},
"ChangeSetName": "ExampleChangeSet",
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
"StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/
ab664180-f686-11ee-9e29-12cd92393671",

```

Utilisation d'ensembles de modifications sensibles à la dérive

Les ensembles de modifications sensibles à la dérive sont des ensembles de CloudFormation modifications améliorés qui vous permettent d'identifier et de gérer la dérive des piles de manière

sûre. Les piles s'écartent de la configuration de modèle attendue lorsque les ressources des piles sont modifiées à l'extérieur CloudFormation, à l'aide du AWS Management Console SDK ou du SDK des services sous-jacents. AWS CLI Les ensembles de modifications sensibles à la dérive comparent les modèles à l'état réel de vos ressources de pile et alignent les ressources dérivées sur leurs définitions de modèles. Lorsque la définition du modèle d'une ressource est mise à jour pour correspondre à son état réel, les ensembles de modifications sensibles à la dérive réinitialisent le statut de dérive de la ressource sans modifier la ressource.

Les avantages de l'utilisation d'ensembles de modifications sensibles à la dérive sont les suivants :

- Prévisualisez les remplacements de dérive : détectez si un déploiement annulera les out-of-band modifications critiques apportées lors de la réponse à un incident.
- Réconciliation systématique des dérives : alignez en toute sécurité les ressources dérivées avec les définitions de modèles conformes, soit en remplaçant les propriétés dérivées par les valeurs du modèle, soit en mettant à jour le modèle pour qu'il corresponde à l'état réel des ressources.
- Fonctionnalités de restauration améliorées : en cas d'échec d'un déploiement, restaurez les ressources dans leur état actuel d'avant le déploiement, et non dans l'état du modèle précédent.
- Visibilité complète — Découvrez exactement comment votre déploiement affectera les configurations réelles des ressources avant d'apporter des modifications.

Les ensembles de modifications traditionnels fournissent une comparaison de votre nouveau modèle avec le modèle précédent pour une pile, mais ne tiennent pas compte de la dérive de la pile. Les ensembles de modifications sensibles à la dérive résolvent ce problème en fournissant une comparaison à trois niveaux entre :

- État réel : configuration en temps réel de vos ressources. CloudFormation lira la configuration des ressources de votre compte au moment de la création de l'ensemble de modifications via le service sous-jacent APIs.
- État de déploiement précédent : configuration définie dans le modèle de votre dernier CloudFormation déploiement.
- État souhaité : configuration définie dans votre nouveau modèle.

Les ensembles de modifications sensibles à la dérive mettront à jour l'état réel de toutes les ressources de la pile pour qu'il corresponde à l'état souhaité, même si aucune ressource n'a été explicitement modifiée dans le modèle.

Considérations

- **Prise en charge des types de ressources** : les ensembles de modifications sensibles à la dérive permettent de comparer l'état souhaité avec l'état réel pour des centaines de types de ressources. Pour les types de ressources non pris en charge, les ensembles de modifications sensibles à la dérive se basent sur la comparaison de l'état de déploiement précédent avec l'état souhaité. Consultez la section [Types de ressources pris en charge](#) pour plus de détails.
- **Propriétés en écriture seule** : pour les propriétés contenant des données sensibles (mots de passe, secrets), les ensembles de modifications sensibles à la dérive sont comparés aux valeurs de déploiement précédentes plutôt qu'aux valeurs réelles.
- **AWS-propriétés gérées** : les ensembles de modifications sensibles à la dérive préservent les out-of-band modifications apportées par les AWS services aux propriétés gérées, telles que la capacité souhaitée d'un groupe Auto Scaling. Consultez la section [des propriétés AWS gérées](#) pour plus de détails.
- **Clés de balise externes** : les ensembles de modifications sensibles à la dérive ne suppriment ni ne modifient les clés de balise qui n'ont pas été spécifiées dans le modèle, évitant ainsi les conflits avec les systèmes de contrôle d'accès basé sur les attributs (ABAC).
- **Remplacement des ressources dérivées** : les ensembles de modifications sensibles à la dérive ne prennent pas en charge la réconciliation par dérive pour les propriétés immuables.
- **Pièces jointes croisées** : certains types de ressources, tels que `AWS::IAM::Policy`, peuvent influencer l'état réel d'autres types de ressources, tels que `AWS::IAM::Role`. Les ensembles de modifications sensibles à la dérive gèrent l'attachement des ressources au sein d'une pile. Si une ressource est modifiée par l'attachement d'une ressource secondaire provenant d'une pile différente, les ensembles de modifications sensibles à la dérive détecteront la modification comme une dérive et pourront annuler l'attachement. Les ressources joignables les plus populaires incluent `AWS::IAM::Policy`, `AWS::IAM::ManagedPolicy`, `AWS::EC2::SecurityGroupIngress`, et `AWS::EC2::SecurityGroupEgress`.

AWS-propriétés gérées

Vous pouvez configurer des propriétés de ressources spécifiques pour une AWS gestion active. Par exemple, vous pouvez autoriser Amazon Relational Database Service (Amazon RDS) à mettre à niveau automatiquement la version mineure du moteur d'une table Amazon RDS. Ces modifications peuvent se traduire par une dérive de la pile CloudFormation. Les ensembles de modifications sensibles à la dérive reconnaissent que la dérive est attendue pour les propriétés AWS gérées et

ne modifient pas leur valeur réelle si vous n'avez pas modifié la propriété dans leur modèle. Les principaux exemples de propriétés AWS gérées sont les suivants :

- Activation de la `AutoMinorVersionUpgrade` propriété d'une table Amazon RDS pour autoriser les mises à jour automatiques de la version du moteur.
- Utilisation de la `AWS::ApplicationAutoScaling::ScalableTarget` ressource pour activer l'auto-scaling pour des propriétés telles que les unités de read/write capacité d'une table Amazon DynamoDB et le nombre souhaité d'un cluster Amazon Elastic Container Service.
- En utilisant les groupes `AWS::AutoScaling::ScalingPolicy for Auto Scaling`.

Les ensembles de modifications sensibles à la dérive clarifient les propriétés identifiées comme gérées. AWS Consultez la AWS CLI section pour plus de détails.

Utilisation d'ensembles de modifications sensibles à la dérive (console)

Vous pouvez créer et gérer des ensembles de modifications sensibles à la dérive via la CloudFormation console en utilisant le même flux de travail que les ensembles de modifications traditionnels, avec des options supplémentaires pour les modes de déploiement.

Création d'ensembles de modifications sensibles à la dérive

Pour créer un ensemble de modifications adapté à la dérive (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Stacks, sélectionnez la pile que vous souhaitez mettre à jour.
3. Choisissez Stack actions, puis choisissez Create change set for current stack.
4. Sur la page Créer un ensemble de modifications, pour Modifier le type de jeu, choisissez Drift aware change set.
5. Choisissez Create change set (Créer un jeu de modifications).
6. Sur la page des détails de l'ensemble de modifications, passez en revue la comparaison tridimensionnelle indiquant le déploiement réel, le déploiement précédent et l'état souhaité pour chaque ressource affectée.
7. Si vous êtes satisfait des modifications, choisissez Execute change set.

Révision des ensembles de modifications sensibles à la dérive

Lorsque vous consultez un ensemble de modifications sensible à la dérive dans la console, vous obtenez des informations améliorées par rapport aux ensembles de modifications traditionnels :

- État de dérive de la pile : indique si la pile a dérivé depuis son dernier déploiement.
- Comparaison des propriétés — Affiche une différence JSON entre l'état réel et l'état souhaité d'une ressource affectée.
- Indicateurs de dérive : indique clairement les propriétés qui ont dérivé dans le différentiel JSON. Cliquez sur Afficher la dérive pour voir la valeur de déploiement précédente d'une propriété.
- Indicateurs de source de valeur : indique si les valeurs antérieures à une propriété proviennent de l'état réel ou de l'état de déploiement précédent.

Utilisation d'ensembles de modifications sensibles à la dérive (AWS CLI)

Vous pouvez créer et gérer des ensembles de modifications sensibles à la dérive à l'aide du en AWS CLI ajoutant le `--deployment-mode REVERT_DRIFT` paramètre à la commande. `create-change-set`

Création d'ensembles de modifications sensibles à la dérive

Pour créer un ensemble de modifications adapté à la dérive

Utilisez la [create-change-set](#) commande avec le `--deployment-mode REVERT_DRIFT` paramètre pour créer un ensemble de modifications adapté à la dérive.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name my-drift-aware-changeset \  
  --template-body file://updated-template.yaml \  
  --deployment-mode REVERT_DRIFT \  
  --capabilities CAPABILITY_IAM
```

Révision des ensembles de modifications sensibles à la dérive

Pour consulter les détails d'un ensemble de modifications sensibles à la dérive

Utilisez la [describe-change-set](#) commande pour passer en revue les détails d'un ensemble de modifications adapté à la dérive :

```
aws cloudformation describe-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

La réponse inclut des informations améliorées pour les ensembles de modifications sensibles à la dérive :

- **StackDriftStatus**— Indique si la pile a dérivé (DRIFTED, IN_SYNCNOT_CHECKED, ouUNKNOWN).
- **ResourceDriftStatus**— Affiche l'état de dérive pour chaque ressource (DELETEDMODIFIED,IN_SYNC, ouNOT_CHECKED).
- **BeforeValueFrom**— Indique si la valeur avant d'une propriété provient de ACTUAL_STATE ouPREVIOUS_DEPLOYMENT_STATE.
- **Drift**— Contient les détails de dérive d'une propriétéPreviousValue, notammentActualValue, etDriftDetectionTimestamp.
- **ResourceDriftIgnoredProperties**— Contient les propriétés d'une ressource pour laquelle l'ensemble de modifications n'annulera pas la dérive et les raisons pour lesquelles la dérive n'est pas prise en compte.

Pour plus d'informations, consultez [DescribeChangeSet](#) dans la Référence d'API AWS CloudFormation .

Exécution d'ensembles de modifications sensibles à la dérive

Pour exécuter un ensemble de modifications adapté à la dérive

Après avoir examiné l'ensemble de modifications, utilisez la [execute-change-set](#)commande pour appliquer les modifications :

```
aws cloudformation execute-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

En cas d'échec du déploiement, les ressources CloudFormation seront ramenées à leur état actuel d'avant le déploiement et préservera out-of-band les modifications apportées depuis le dernier déploiement.

Types de ressources pris en charge

Les ensembles de modifications sensibles à la dérive permettent une comparaison tridimensionnelle de l'état réel, de l'état de déploiement précédent et de l'état souhaité pour tous les types de [ressources prenant en charge la détection de la dérive, à l'exception des types](#) de ressources suivants :

- `AWS::SageMaker::MlflowTrackingServer`
- `AWS::Route53Resolver::FirewallRuleGroup`
- `AWS::MediaLive::Multiplexprogram`
- `AWS::VpcLattice::ResourceConfiguration`
- `AWS::S3::StorageLensGroup`
- `AWS::Bedrock::AgentAlias`
- `AWS::MSK::Cluster`
- `AWS::RDS::DBProxy`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::QBusiness::Index`
- `AWS::NetworkManager::CoreNetwork`
- `AWS::IAM::OIDCProvider`
- `AWS::Organizations::ResourcePolicy`
- `AWS::SNS::TopicInlinePolicy`
- `AWS::Route53::KeySigningKey`
- `AWS::DataZone::PolicyGrant`
- `AWS::Transfer::Certificate`
- `AWS::SageMaker::ImageVersion`
- `AWS::Neptune::DBParameterGroup`
- `AWS::ODB::CloudVmCluster`
- `AWS::RolesAnywhere::TrustAnchor`
- `AWS::Detective::Graph`
- `AWS::Maester::DocumentType`

- `AWS::SageMaker::ModelPackageGroup`
- `AWS::S3Express::BucketPolicy`
- `AWS::Panorama::PackageVersion`
- `AWS::S3Tables::TableBucketPolicy`

Les ensembles de modifications sensibles à la dérive se basent sur une comparaison de l'état de déploiement précédent et de l'état souhaité pour les ressources qui ne prennent pas en charge la comparaison tridimensionnelle.

Exécuter un ensemble de modifications pour une CloudFormation pile

Pour appliquer les modifications décrites dans un jeu de modifications à votre pile, exécutez ce dernier.

Important

Après avoir exécuté un ensemble de modifications, CloudFormation supprime tous les ensembles de modifications supplémentaires associés à la pile car ils ne sont plus valides pour la pile mise à jour. En cas d'échec de mise à jour, vous devez créer un autre jeu de modifications.

Politiques de piles et exécution d'un jeu de modifications

Si vous exécutez un jeu de modifications au niveau d'une pile qui est associée à une stratégie, CloudFormation applique cette stratégie lorsqu'il met à jour cette pile. Lorsque vous exécutez un jeu de modifications, vous ne pouvez pas spécifier une politique de pile temporaire afin de remplacer la politique existante. Pour mettre à jour une ressource protégée, vous devez mettre à jour la politique de la pile ou utiliser la méthode de mise à jour directe. Pour de plus amples informations, veuillez consulter [Mettre à jour directement les piles](#).

Exécute a change set (console)

Pour exécuter un jeu de modifications

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.

3. Sur la page Piles, sélectionnez le nom de la pile que vous voulez mettre à jour.
4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Choisissez le nom de l'ensemble de modifications que vous souhaitez exécuter.
6. Sur la page de détails du jeu de modifications, sélectionnez Exécuter le jeu de modifications.

CloudFormation commence immédiatement à mettre à jour la pile. La CloudFormation console vous dirige vers l'onglet Events, où vous pouvez suivre la progression de la mise à jour de la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Exécute a change set for nested stacks (console)

Pour exécuter un jeu de modifications pour des piles imbriquées

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez le nom de la pile que vous voulez mettre à jour. Vous devez choisir le nom de la pile associé à l'ensemble de modifications racine.
4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Choisissez le nom de l'ensemble de modifications racine que vous souhaitez exécuter.
6. Sur la page de détails du jeu de modifications, sélectionnez Exécuter le jeu de modifications.

Note

CloudFormation exécute les modifications décrites dans votre ensemble de modifications racine et dans vos ensembles de modifications imbriqués, si l'option **Activé pour les ensembles de modifications pour les piles imbriquées** a été sélectionnée au cours du processus. [Création d'un ensemble de modifications pour une CloudFormation pile](#)

CloudFormation commence immédiatement à mettre à jour la pile. La CloudFormation console vous dirige vers l'onglet Events, où vous pouvez suivre la progression de la mise à

jour de la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Pour exécuter un ensemble de modifications (AWS CLI)

- Exécutez la commande [execute-change-set](#).

Spécifiez l'ID du jeu de modifications que vous souhaitez exécuter, comme illustré dans l'exemple suivant :

```
aws cloudformation execute-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

La commande de cet exemple exécute un jeu de modifications avec l'ID `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000`.

Après avoir exécuté la commande, CloudFormation commence à mettre à jour la pile. Pour afficher la progression de la pile, utilisez la commande [describe-stacks](#).

Supprimer un ensemble de modifications pour une CloudFormation pile

La suppression d'un jeu de modifications entraîne sa suppression de la liste des jeux de modifications de la pile. La suppression d'un jeu de modifications empêche à quiconque de l'exécuter par erreur. À moins que vous ne les supprimiez, CloudFormation conserve tous les ensembles de modifications jusqu'à ce que vous mettiez à jour la pile.

Delete a change set

Pour supprimer un jeu de modifications (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez le nom de la pile contenant le jeu de modifications que vous voulez supprimer.

4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Sélectionnez le nom de l'ensemble de modifications que vous souhaitez supprimer.
6. Sur la page de détails du jeu de modifications, sélectionnez Supprimer le jeu de modifications.

CloudFormation commence immédiatement à supprimer l'ensemble de modifications de la liste des ensembles de modifications de la pile, et vous êtes redirigé vers la page Stacks.

Delete a change set for nested stacks (console)

Pour supprimer un jeu de modifications pour des piles imbriquées

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez le nom de la pile associée à l'ensemble de modifications racine.
4. Dans le volet de navigation, choisissez Jeux de modifications pour afficher la liste des jeux de modifications de la pile.
5. Sélectionnez le nom de l'ensemble de modifications que vous souhaitez supprimer.
6. Sur la page des détails du jeu de modifications, choisissez Supprimer. En choisissant Supprimer le jeu de modifications, vous supprimerez toute la hiérarchie des jeux de modifications imbriqués.

Note

L'opération de suppression des ensembles de modifications pour les piles imbriquées est asynchrone et affiche un statut DELETE_PENDING, suivi d'un statut DELETE_IN_PROGRESS. Une fois l'opération de suppression de l'ensemble de modifications terminée, les ensembles de modifications seront retirés de la liste. Les piles imbriquées dans le statut REVIEW_IN_PROGRESS seront également supprimées si elles ont été créées lors de la création de l'ensemble de modifications.

CloudFormation commence immédiatement à supprimer l'ensemble de modifications de la liste des ensembles de modifications de la pile.

 Note

Si des piles imbriquées restent bloquées sur l'état « en cours », consultez la page [Dépannage des erreurs](#) dans le document [Échec de la restauration des piles imbriquées](#).

Pour supprimer un ensemble de modifications (AWS CLI)

- Exécutez la commande [delete-change-set](#) en spécifiant l'ID du jeu de modifications que vous souhaitez supprimer, comme illustré dans l'exemple suivant :

```
aws cloudformation delete-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Exemples d'ensembles de modifications pour les CloudFormation piles

Cette section fournit des exemples d'ensembles de modifications que CloudFormation pourraient être créés pour les modifications de pile courantes. Ces exemples illustrent comment modifier un modèle directement, mettre à jour un paramètre d'entrée unique, planifier la recreation (ou le remplacement) des ressources (qui empêche de perdre les données qui n'étaient pas sauvegardées ou d'interrompre des applications en cours d'exécution dans votre pile) et comment ajouter ou supprimer des ressources. Pour décrire le fonctionnement des jeux de modifications, nous parcourons les modifications qui ont été soumises et parlerons du jeu de modifications généré. Comme chaque exemple repose sur le précédent et présume que vous l'avez compris, nous vous recommandons de les lire dans l'ordre. Pour obtenir une description de chaque champ d'un jeu de modifications, veuillez consulter la rubrique [Change](#) dans la Référence d'API AWS CloudFormation .

Vous pouvez utiliser la [console](#) ou l' AWS CLI opération CloudFormation [DescribeChangeSet](#)API pour afficher les détails de l'ensemble de modifications.

Nous avons généré chacun des jeux de modifications suivants à partir d'une pile avec l'[exemple de modèle](#) :

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "A sample EC2 instance template for testing change sets.",
  "Parameters" : {
    "Purpose" : {
      "Type" : "String",
      "Default" : "testing",
      "AllowedValues" : ["testing", "production"],
      "Description" : "The purpose of this instance."
    },
    "KeyPairName" : {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
instance"
    },
    "InstanceType" : {
      "Type" : "String",
      "Default" : "t2.micro",
      "AllowedValues" : ["t2.micro", "t2.small", "t2.medium"],
      "Description" : "The EC2 instance type."
    }
  },
  "Resources" : {
    "MyEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyPairName" },
        "InstanceType" : { "Ref" : "InstanceType" },
        "ImageId" : "ami-8fcee4e5",
        "Tags" : [
          {
            "Key" : "Purpose",
            "Value" : { "Ref" : "Purpose" }
          }
        ]
      }
    }
  }
}
```

Modification directe d'un modèle

Lorsque vous modifiez directement des ressources dans le modèle de la pile afin de générer un jeu de modifications, CloudFormation considère qu'il s'agit d'une modification directe, par opposition aux modifications lancées par une valeur de paramètre mise à jour. Le jeu de modifications suivant, qui a ajouté une balise à l'instance `i-1abc23d4`, est un exemple de modification directe. Toutes les autres valeurs d'entrée, telles que les valeurs de paramètres et les capacités, restent les mêmes. Nous nous concentrerons donc sur la structure `Changes`.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Static",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          }
        ]
      }
    }
  ],
```

```

        "Action": "Modify",
        "Scope": [
            "Tags"
        ],
        "LogicalResourceId": "MyEC2Instance",
        "Replacement": "False"
    },
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T23:35:25.813Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Dans `Changes`, il n'y a qu'une seule structure `ResourceChange`. Cette structure décrit des informations telles que le type de ressource qui CloudFormation va changer, l'action CloudFormation à entreprendre, l'ID de la ressource, l'étendue de la modification et si la modification nécessite un remplacement (CloudFormation création d'une nouvelle ressource puis suppression de l'ancienne). Dans l'exemple, l'ensemble de modifications indique que cela CloudFormation modifiera l'attribut `Tags` de l'instance EC2 `i-1abc23d4` et ne nécessite pas le remplacement de l'instance.

Dans la `Details` structure, CloudFormation qualifie cette modification de modification directe qui ne nécessitera jamais de recréer (remplacer) l'instance. Vous pouvez exécuter cette modification en toute confiance, en sachant que cela CloudFormation ne remplacera pas l'instance.

CloudFormation montre ce changement sous forme d'évaluation statique. Une évaluation statique permet de déterminer la valeur de la balise avant d'exécuter l'ensemble de modifications.

Dans certains cas, vous ne pouvez pas déterminer une valeur qu'après avoir exécuté un ensemble de modifications. CloudFormation qualifie ces modifications d'évaluations dynamiques. Par exemple, si vous faites référence à une ressource mise à jour qui est remplacée de manière conditionnelle, vous ne pouvez pas déterminer si la référence à la ressource mise à jour va changer.

Modification d'une valeur de paramètre d'entrée

Lorsque vous modifiez une valeur de paramètre d'entrée, CloudFormation génère deux modifications pour chaque ressource qui utilise la valeur de paramètre mise à jour. Dans cet exemple, nous

souhaitons mettre en évidence ce à quoi ressemblent ces modifications, ainsi que les informations que vous devez cibler. L'exemple suivant a été généré en modifiant uniquement la valeur du paramètre d'entrée Purpose.

Le Purpose paramètre spécifie une valeur de clé de balise pour l' EC2 instance. Dans cet exemple, la valeur de paramètre testing a été remplacée par production. La nouvelle valeur apparaît dans la structure Parameters.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet",
  "Parameters": [
    {
      "ParameterValue": "production",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Dynamic",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          }
        ],
        "CausingEntity": "Purpose",
```

```

        "ChangeSource": "ParameterReference",
        "Evaluation": "Static",
        "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never"
        }
    },
    "Action": "Modify",
    "Scope": [
        "Tags"
    ],
    "LogicalResourceId": "MyEC2Instance",
    "Replacement": "False"
},
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T23:59:18.447Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

La structure `Changes` fonctionne de manière similaire à l'exemple [Modification directe d'un modèle](#). Il n'existe qu'une seule `ResourceChange` structure ; elle décrit une modification de l'attribut `Tags` de l'i-1abc23d4 EC2 instance.

Toutefois, dans la structure `Details`, le jeu de modifications présente deux changements pour l'attribut `Tags`, même si une seule valeur de paramètre a été modifiée. Les ressources qui référencent une valeur de paramètre modifiée (à l'aide de la fonction intrinsèque `Ref`) entraînent toujours deux modifications : une avec une évaluation `Dynamic` et une autre avec une évaluation `Static`. Pour voir ces types de modification, consultez les champs suivants :

- Pour la modification de l'évaluation `Static`, consultez le champ `ChangeSource`. Dans cet exemple, le champ `ChangeSource` équivaut à `ParameterReference`, ce qui signifie que cette modification est le résultat de la mise à jour de la valeur de référence du paramètre. Le jeu de modifications doit comporter une modification de l'évaluation `Dynamic` similaire.

- Pour identifier la modification de l'évaluation Dynamic correspondante, comparez la structure `Target` des deux modifications, qui contient les mêmes informations. Dans cet exemple, la structure `Target` des deux modifications contient les mêmes valeurs pour les champs `Attribute` et `RequireRecreation`.

Pour ces types de modifications, concentrez-vous sur l'évaluation statique, qui fournit des informations plus détaillées sur la modification. Dans cet exemple, l'évaluation statique indique que le changement est le résultat de la modification de la valeur de référence du paramètre (`ParameterReference`). Le paramètre exact qui a été modifié est signalé par le champ `CauseEntity` (le paramètre `Purpose`).

Détermination de la valeur du champ `Replacement`

Le `Replacement` champ d'une `ResourceChange` structure indique s'il est CloudFormation prévu de recréer la ressource. La planification de la recréation (ou du remplacement) de la ressource vous empêche de perdre les données qui n'étaient pas sauvegardées ou d'interrompre des applications en cours d'exécution dans votre pile.

La valeur du champ `Replacement` varie selon qu'une modification nécessite un remplacement, ce qui est indiqué par le champ `RequiresRecreation` dans la structure `Target` d'une modification. Par exemple, si le champ `RequiresRecreation` indique `Never`, le champ `Replacement` contient la valeur `False`. Toutefois, si plusieurs modifications sont apportées à une seule ressource et que chaque modification a une valeur différente pour le `RequiresRecreation` champ, CloudFormation met à jour la ressource en utilisant le comportement le plus intrusif. En d'autres termes, si seulement l'une des nombreuses modifications nécessite un remplacement, CloudFormation doit remplacer la ressource et, par conséquent, définir le champ `Replacement` sur `True`.

L'ensemble de modifications suivant a été généré en modifiant les valeurs de chaque paramètre (`PurposeInstanceType`, et `KeyPairName`), qui sont tous utilisés par l' EC2 instance. Avec ces modifications, il CloudFormation sera nécessaire de remplacer l'instance car le `Replacement` champ est égal à `True`.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-multiple",
  "Parameters": [
    {
```

```

        "ParameterValue": "production",
        "ParameterKey": "Purpose"
    },
    {
        "ParameterValue": "MyNewKeyName",
        "ParameterKey": "KeyPairName"
    },
    {
        "ParameterValue": "t2.small",
        "ParameterKey": "InstanceType"
    }
],
"Changes": [
    {
        "ResourceChange": {
            "ResourceType": "AWS::EC2::Instance",
            "PhysicalResourceId": "i-7bef86f8",
            "Details": [
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Properties",
                        "Name": "KeyName",
                        "RequiresRecreation": "Always"
                    }
                },
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Properties",
                        "Name": "InstanceType",
                        "RequiresRecreation": "Conditionally"
                    }
                },
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never"
                    }
                }
            ]
        }
    },

```

```

        {
            "CausingEntity": "KeyPairName",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Properties",
                "Name": "KeyName",
                "RequiresRecreation": "Always"
            }
        },
        {
            "CausingEntity": "InstanceType",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Properties",
                "Name": "InstanceType",
                "RequiresRecreation": "Conditionally"
            }
        },
        {
            "CausingEntity": "Purpose",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Tags",
                "RequiresRecreation": "Never"
            }
        }
    ],
    "Action": "Modify",
    "Scope": [
        "Tags",
        "Properties"
    ],
    "LogicalResourceId": "MyEC2Instance",
    "Replacement": "True"
},
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T00:39:35.974Z",
"Capabilities": [],
"StackName": "MyStack",

```

```

    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-multiple/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Consultez chaque modification (évaluations statiques dans la structure `Details`) pour identifier celle qui nécessite le remplacement de la ressource. Dans cet exemple, chaque modification a une valeur différente pour le `RequireRecreation` champ, mais la modification apportée à la `KeyName` propriété présente le comportement de mise à jour le plus intrusif, nécessitant toujours une recréation. CloudFormation remplacera l'instance car le nom de la clé a été modifié.

Si le nom de clé n'avait pas changé, la modification de la propriété `InstanceType` aurait le comportement de mise à jour le plus intrusif (`Conditionally`). Le champ `Replacement` indiquerait `Conditionally`. Pour connaître les conditions dans lesquelles l' CloudFormation instance est remplacée, consultez le comportement de mise à jour de la `InstanceType` propriété du type de [AWS::EC2::Instance](#) ressource.

Ajout et suppression de ressources

L'exemple suivant a été généré en soumettant un modèle modifié qui supprime l' EC2 instance et ajoute un groupe Auto Scaling et une configuration de lancement.

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-addremove",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [

```

```

{
  "ResourceChange": {
    "Action": "Add",
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
    "Scope": [],
    "Details": [],
    "LogicalResourceId": "AutoScalingGroup"
  },
  "Type": "Resource"
},
{
  "ResourceChange": {
    "Action": "Add",
    "ResourceType": "AWS::AutoScaling::LaunchConfiguration",
    "Scope": [],
    "Details": [],
    "LogicalResourceId": "LaunchConfig"
  },
  "Type": "Resource"
},
{
  "ResourceChange": {
    "ResourceType": "AWS::EC2::Instance",
    "PhysicalResourceId": "i-1abc23d4",
    "Details": [],
    "Action": "Remove",
    "Scope": [],
    "LogicalResourceId": "MyEC2Instance"
  },
  "Type": "Resource"
}
],
"CreationTime": "2020-11-18T01:44:08.444Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-addremove/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Changes comporte trois structures ResourceChange, une pour chaque ressource. Pour chaque ressource, le Action champ indique si elle est CloudFormation ajoutée ou supprimée. Les champs Scope et Details sont vides, car ils s'appliquent uniquement aux ressources modifiées.

Pour les nouvelles ressources, CloudFormation vous ne pouvez pas déterminer la valeur de certains champs tant que vous n'avez pas exécuté l'ensemble de modifications. Par exemple, CloudFormation ne fournit pas le physique IDs du groupe Auto Scaling ni la configuration de lancement car ils n'existent pas encore. CloudFormation crée les nouvelles ressources lorsque vous exécutez l'ensemble de modifications.

Affichage des modifications au niveau des propriétés

L'exemple suivant montre les modifications apportées au niveau de la propriété à la Tag propriété d'une instance Amazon EC2 . Les balises Value et Key seront remplacées par Test.

```
"ChangeSetName": "SampleChangeSet",
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/38d91d27-798d-4736-9bf1-fb7c46207807",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
SampleEc2Template/68edcdc0-f6b6-11ee-966c-126d572cdd11",
  "StackName": "SampleEc2Template",
  "Description": "A sample EC2 instance template for testing change sets.",
  "Parameters": [
    {
      "ParameterKey": "KeyPairName",
      "ParameterValue": "BatchTest"
    },
    {
      "ParameterKey": "Purpose",
      "ParameterValue": "testing"
    },
    {
      "ParameterKey": "InstanceType",
      "ParameterValue": "t2.micro"
    }
  ],
  "CreationTime": "2024-04-09T21:29:10.759000+00:00",
  "ExecutionStatus": "AVAILABLE",
  "Status": "CREATE_COMPLETE",
  "StatusReason": null,
  "NotificationARNs": [],
  "RollbackConfiguration": {
    "...skipping..."
  },
  "Changes": [
    {
      "Type": "Resource",
```



```

    "ResourceChange": {
      "Action": "Modify",
      "LogicalResourceId": "MyEC2Instance",
      "PhysicalResourceId": "i-0cc7856a36315e62b",
      "ResourceType": "AWS::EC2::Instance",
      "Replacement": "False",
      "Scope": [
        "Tags"
      ],
      "Details": [
        {
          "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never",
            "Path": "/Properties/Tags/0/Value",
            "BeforeValue": "testing",
            "AfterValue": "Test",
            "AttributeChangeType": "Modify"
          },
          "Evaluation": "Static",
          "ChangeSource": "DirectModification"
        },
        {
          "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never",
            "Path": "/Properties/Tags/0/Key",
            "BeforeValue": "Purpose",
            "AfterValue": "Test",
            "AttributeChangeType": "Modify"
          },
          "Evaluation": "Static",
          "ChangeSource": "DirectModification"
        }
      ],
      "BeforeContext": "{\n\"Properties\":{\n\"KeyName\":\n\"BatchTest\", \n\"ImageId\n\":\n\"ami-8fcee4e5\", \n\"InstanceType\":\n\"t2.micro\", \n\"Tags\":[\n{\n\"Value\":\n\"testing\", \n\"Key\":\n\"Purpose\"}\n]}\n}",
      "AfterContext": "{\n\"Properties\":{\n\"KeyName\":\n\"BatchTest\", \n\"ImageId\n\":\n\"ami-8fcee4e5\", \n\"InstanceType\":\n\"t2.micro\", \n\"Tags\":[\n{\n\"Value\":\n\"Test\", \n\"Key\n\":\n\"Test\"}\n]}\n}"
    }
  }
}

```

]

La structure `Details` affiche les valeurs de `Key` et `Value` avant l'exécution du jeu de modifications, ainsi que celles après l'exécution du jeu de modifications.

Modifier les ensembles pour les piles imbriquées

Avec les ensembles de modifications pour les piles imbriquées, vous pouvez prévisualiser les modifications apportées à vos ressources d'application et d'infrastructure sur l'ensemble de la hiérarchie de pile imbriquée et procéder aux mises à jour lorsque vous avez confirmé que toutes les modifications sont celles que vous comptiez faire.

Pour plus de détails sur les ensembles de modifications pour les piles imbriquées, reportez-vous aux sections suivantes :

Rubriques

- [Présentation des ensembles de modifications et des piles imbriquées](#)
- [Travailler avec des ensembles de modifications pour les piles imbriquées \(console\)](#)
- [Utilisation des ensembles de modifications pour les piles imbriquées \(AWS CLI\)](#)

Présentation des ensembles de modifications et des piles imbriquées

Les ensembles de modifications pour les piles imbriquées combinent les fonctions suivantes ensemble pour étendre la portée de la prévisualisation des modifications à l'ensemble de la hiérarchie de pile :

- Un ensemble de modifications est une CloudFormation fonctionnalité qui offre un aperçu de l'impact des modifications proposées sur une pile sur les ressources existantes ou nouvellement créées. Lors de la création d'un ensemble de modifications, CloudFormation fournit une liste des modifications proposées en comparant votre pile aux modifications apportées aux ressources que vous avez soumises. Pour plus d'informations sur les jeux de modifications, consultez [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).
- Une pile imbriquée est une pile créée dans le cadre d'une autre pile. Par exemple, vous pouvez disposer de ressources liées à la sécurité ou à la mise en réseau dans une pile imbriquée et de ressources applicatives dans une autre. Le partitionnement des modèles d'application de cette façon aide à entretenir et réutiliser le code. Pour plus d'informations sur les piles imbriquées, consultez [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).

Travailler avec des ensembles de modifications pour les piles imbriquées (console)

- Créer un ensemble de modifications – crée un ensemble de modifications en soumettant des modifications à partir de n'importe quel niveau de la hiérarchie de pile. Vous pouvez soumettre un modèle de pile modifié ou des valeurs de paramètre d'entrée modifiées et CloudFormation compare votre pile imbriquée aux modifications que vous avez soumises pour générer un ensemble de modifications. Les ensembles de modifications pour les piles imbriquées sont activés par défaut dans la console CloudFormation . Pour de plus amples informations, veuillez consulter [Création d'un ensemble de modifications pour une CloudFormation pile](#).

▼ Change sets

Change sets for nested stacks

Creates change sets for all nested stacks specified in your template. [Learn more](#) 

☒ **Enabled**

Creates a change set for all nested stacks specified in the template.

☐ **Disabled**

Create a change set only on the target stack.

Note

L'ensemble de modifications racine est l'ensemble de modifications associé à la pile à partir de laquelle toute la hiérarchie des ensembles de modifications est créée. Vous devez exécuter ou supprimer des ensembles de modifications pour les piles imbriquées de l'ensemble de modifications racine. Pour de plus amples informations, veuillez consulter [Exécution d'opérations sur des piles imbriquées](#).

- Afficher l'ensemble de modifications – Visualisez les modifications apportées aux ressources dans les piles imbriquées avant de les exécuter. Vous pouvez afficher les modifications proposées dans la section Modifications de votre ensemble de modifications en naviguant dans la pile active et ses ensembles de modifications imbriqués. Pour de plus amples informations, veuillez consulter [Afficher un ensemble de modifications pour une CloudFormation pile](#).
- Exécuter l'ensemble de modifications – Exécutez les modifications décrites dans l'ensemble de modifications qui concerne la pile active et ses descendants. L'opération d'exécution doit être effectuée à partir de l'ensemble de modifications racine. Pour de plus amples informations, veuillez consulter [Exécuter un ensemble de modifications pour une CloudFormation pile](#).
- Supprimer l'ensemble de modifications – supprime les ensembles de modifications de la pile active. La suppression d'un ensemble de modifications vous empêche, vous ou un autre utilisateur, d'initier accidentellement un ensemble de modifications qui ne doit pas être appliqué. L'opération

de suppression doit être exécutée à partir de l'ensemble de modifications racine. Pour de plus amples informations, veuillez consulter [Supprimer un ensemble de modifications pour une CloudFormation pile](#).

Utilisation des ensembles de modifications pour les piles imbriquées (AWS CLI)

create-change-set

- [create-change-set](#) : les ensembles de modifications pour les piles imbriquées ne sont pas activés par défaut pour la AWS CLI. Pour créer un jeu de modifications pour l'ensemble de la hiérarchie de piles, spécifiez l'option `--include-nested-stacks`. Pour de plus amples informations, veuillez consulter [Création d'un ensemble de modifications pour une CloudFormation pile](#).

L' AWS CLI exemple suivant crée un ensemble de modifications pour la pile racine spécifiée.

```
aws cloudformation create-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM \  
  --include-nested-stacks
```

Voici un exemple de sortie.

```
{  
  "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:Stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204"  
}
```

describe-change-set

- [describe-change-set](#)— Renvoie la liste des modifications qui CloudFormation seront apportées si vous exécutez l'ensemble de modifications. Si l'ensemble de modifications spécifié contient des ensembles de modifications enfant appartenant à des piles imbriquées, ChangeSetId renverra des informations sur cet ensemble de modifications. Pour de plus amples informations, veuillez consulter [Afficher un ensemble de modifications pour une CloudFormation pile](#).

L' AWS CLI exemple suivant décrit le jeu de modifications pour la pile racine spécifiée.

```
aws cloudformation describe-change-set \  
  --change-set-name my-root-stack-change-set \  
  --stack-name my-root-stack
```

Voici un exemple de sortie.

```
{  
  "Changes": [  
    {  
      "Type": "Resource",  
      "ResourceChange": {  
        "Action": "Modify",  
        "LogicalResourceId": "ChildStack",  
        "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",  
        "ResourceType": "AWS::CloudFormation::Stack",  
        "Replacement": "False",  
        "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",  
        "Scope": [  
          "Properties"  
        ],  
        "Details": [  
          {  
            "Target": {  
              "Attribute": "Properties",  
              "RequiresRecreation": "Never"  
            },  
            "Evaluation": "Dynamic",  
            "ChangeSource": "Automatic"  
          }  
        ]  
      }  
    ]  
  },  
  "ChangeSetName": "my-root-stack-change-set",  
  "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

"StackName": "my-root-stack",
"IncludeNestedStacks": true,
"ParentChangeSetId": null,
"RootChangeSetId": null,
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
    "CAPABILITY_IAM"
],
"Tags": null
}

```

L' AWS CLI exemple suivant décrit le jeu de modifications pour la pile imbriquée spécifiée.

```

aws cloudformation describe-change-set \
  --change-set-name my-nested-stack-change-set \
  --stack-name my-nested-stack

```

Voici un exemple de sortie.

```

{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "function",
        "PhysicalResourceId": "my-function",
        "ResourceType": "AWS::Lambda::Function",
        "Replacement": "False",
        "ChangeSetId": null,
        "Scope": [
          "Properties"

```

```

    ],
    "Details": [
      {
        "Target": {
          "Attribute": "Properties",
          "Name": "Timeout",
          "RequiresRecreation": "Never"
        },
        "Evaluation": "Static",
        "ChangeSource": "DirectModification"
      }
    ]
  }
},
"ChangeSetName": "my-nested-stack-change-set",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
"ParentChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"RootChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"IncludeNestedStacks": true,
"StackName": "my-nested-stack",
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "UNAVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": "Executable from root change set",
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
  "CAPABILITY_IAM"
],
"Tags": null
}

```

execute-change-set

- [execute-change-set](#) – Crée ou met à jour une pile à l'aide des informations d'entrée qui ont été fournies lorsque l'ensemble de modifications spécifié a été créé. Pour créer un jeu de modifications pour toute la hiérarchie de piles, vous devez spécifier l'option `--include-nested-stacks` lors de l'opération `create-change-set`. Pour de plus amples informations, veuillez consulter [Exécuter un ensemble de modifications pour une CloudFormation pile](#).

Note

`execute-change-set` doit être exécuté à partir de l'ensemble de modifications racine et appliquera l'ensemble de modifications à l'ensemble de la hiérarchie des piles.

L' AWS CLI exemple suivant exécute un ensemble de modifications pour la pile racine spécifiée.

```
aws cloudformation execute-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set
```

delete-change-set

- [delete-change-set](#) – Supprime l'ensemble de modifications spécifié. La suppression des ensembles de modifications garantit que personne n'utilise un ensemble de modifications inapproprié. La suppression des jeux de modifications est asynchrone pour les jeux de modifications créés avec l'option `--include-nested-stacks`. Pour de plus amples informations, veuillez consulter [Supprimer un ensemble de modifications pour une CloudFormation pile](#).

Note

`delete-change-set` doit être exécuté à partir de l'ensemble de modifications racine et supprimera toute la hiérarchie des ensembles de modifications. Les piles imbriquées dans l'état `REVIEW_IN_PROGRESS` seront également supprimées si elles ont été créées pendant l'opération `create-change-set`.

L' AWS CLI exemple suivant supprime le jeu de modifications pour la pile racine spécifiée.

```
aws cloudformation delete-change-set \  
  --stack-name my-root-stack
```



```
--stack-name my-root-stack \  
--change-set-name my-root-stack-change-set
```

Valider les déploiements de stack

Grâce à la validation préalable au déploiement, vous pouvez identifier et résoudre les problèmes de déploiement potentiels avant d'exécuter vos ensembles de CloudFormation modifications. Cette fonctionnalité valide vos modèles par rapport aux scénarios de défaillance courants, vous aidant ainsi à détecter les problèmes dès le début du cycle de développement.

Rubriques

- [Comment fonctionne la validation préalable au déploiement](#)
- [Considérations](#)
- [Conditions préalables](#)
- [Valider un déploiement de stack \(console\)](#)
- [Valider un déploiement de stack \(AWS CLI\)](#)
- [Types de validation](#)
- [Limite de ressources](#)

Comment fonctionne la validation préalable au déploiement

La validation préalable au déploiement comprend les phases suivantes :

1. Créez votre ensemble de modifications : générez un ensemble de modifications comme vous le feriez normalement pour les mises à jour de votre CloudFormation pile. La validation préalable au déploiement est activée par défaut lors de la création de votre ensemble de modifications.
2. Exécution de la validation : CloudFormation exécute plusieurs contrôles de validation par rapport à votre modèle et à votre environnement cible. Trois types de validation sont actuellement pris en charge : la validation de la syntaxe des propriétés par rapport aux schémas de ressources, la détection des conflits de noms de ressources avec les ressources existantes et la validation du vide du compartiment S3 pour les opérations de suppression.
3. Vérifiez les résultats de validation : CloudFormation fournit des informations détaillées sur tous les problèmes détectés, y compris le chemin précis indiquant l'emplacement du problème dans le modèle, éliminant ainsi le débogage manuel du modèle.

4. Résoudre les problèmes : résolvez les problèmes identifiés en mettant à jour vos modèles ou en résolvant les conflits avant de procéder au déploiement.
5. Exécutez en toute confiance : déployez votre ensemble de modifications en sachant que les scénarios de défaillance courants ont été validés dès le départ.

Considérations

Lorsque vous utilisez la validation préalable au déploiement, gardez à l'esprit les points suivants :

- La validation préalable au déploiement se concentre sur les trois scénarios d'échec de déploiement courants. Cela ne garantit pas le succès de votre déploiement, mais réduit le risque d'échecs courants.
- Les modes de validation se comportent différemment :
 - Le mode FAIL empêche l'exécution des ensembles de modifications lorsque la validation détecte des erreurs, empêchant ainsi le déploiement des modèles problématiques. Cela s'applique aux erreurs de syntaxe des propriétés et aux conflits de dénomination des ressources.
 - Le mode WARN permet à la création d'un ensemble de modifications de réussir malgré les échecs de validation, en fournissant des avertissements que les développeurs peuvent consulter et corriger avant leur exécution. Cela s'applique aux violations de contraintes telles que le vide du compartiment S3, qui peuvent être résolues par une intervention manuelle.
- Les résultats de validation sont liés à l'ensemble de modifications spécifique. Si vous modifiez votre modèle, vous devrez créer un nouvel ensemble de modifications pour obtenir des résultats de validation actualisés.
- La validation du compartiment S3 vérifie uniquement la présence d'objets, et non les politiques de compartiment ou autres contraintes susceptibles d'empêcher la suppression.

Conditions préalables

Pour utiliser la validation préalable au déploiement, vous devez disposer des éléments suivants :

- Les autorisations IAM nécessaires pour créer des ensembles de modifications et lire des ressources dans votre compte. Pour vérifier le vide du compartiment S3, vous avez besoin d'`s3:ListBucket` une autorisation.
- Accès à l' Région AWS endroit où vos piles sont déployées.
- CloudFormation modèles que vous souhaitez valider avant le déploiement.

Valider un déploiement de stack (console)

Utilisez la procédure suivante pour valider le déploiement de votre stack à l'aide de la console.

Pour valider un modèle avant le déploiement

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de votre pile.
3. Sur la page Piles, sélectionnez la pile en cours d'exécution que vous voulez créer un jeu de modifications.
4. Dans le volet des détails de la pile, choisissez Update Stack, puis Create a change set.
5. Sur la *stack-name* page Créer un ensemble de modifications pour, téléchargez votre modèle mis à jour ou spécifiez la source du modèle.
6. Choisissez Next pour passer aux étapes de configuration restantes de l'ensemble de modifications.
7. Si le modèle inclut des ressources IAM, dans Fonctionnalités, sélectionnez Je reconnais que cela CloudFormation pourrait créer des ressources IAM. Les ressources IAM peuvent modifier les autorisations dans votre compte AWS . Consultez ces ressources pour vous assurer que vous autorisez uniquement les actions que vous prévoyez. Pour plus d'informations, voir [Reconnaissance des ressources IAM dans les CloudFormation modèles](#).
8. Sur la page Révision, choisissez Créer un ensemble de modifications.
9. CloudFormation créera l'ensemble de modifications et exécutera des contrôles de validation. Consultez les résultats de la validation dans l'onglet Validation du déploiement.
10. Si la validation est réussie ou si vous êtes satisfait des avertissements, choisissez le set Execute Change pour déployer vos modifications.
11. Si la validation échoue, corrigez les problèmes et créez un nouvel ensemble de modifications pour revalider votre déploiement.

Valider un déploiement de stack (AWS CLI)

Les AWS CLI commandes de validation avant le déploiement sont les suivantes :

- [create-change-set](#) validation automatique lors de la création de l'ensemble de modifications.

- [describe-change-set](#) pour vérifier le statut de l'ensemble de modifications
- [describe-events](#) pour examiner les résultats de validation.

Utilisez la procédure suivante pour valider le déploiement de votre stack à l'aide du AWS CLI.

Pour valider un modèle avant le déploiement

1. Utilisez la commande [create-change-set](#) :

```
aws cloudformation create-change-set \  
  --stack-name MyStack \  
  --change-set-name MyChangeSet \  
  --change-set-type "CREATE" \  
  --template-body file://updated-template.yaml
```

La commande renvoie à la fois l'ARN de l'ensemble de modifications et l'ARN de la pile.

2. Utilisez la commande [describe-events](#) avec l'ARN du jeu de modifications ou le nom du jeu de modifications pour vérifier l'état de validation et les résultats.

```
aws cloudformation describe-events \  
  --change-set-name "arn:aws:cloudformation:us-east-1:123456789012:changeSet/MyChangeSet/94498df5-1afb-43b1-9869-9f82b2d877ac"
```

Exemple de sortie d'une erreur de validation :

```
{  
  "OperationEvents": [  
    {  
      "EventId": "9b5c9a29-4704-4ad0-8082-afb49418d55b",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",  
      "OperationStatus": "FAILED",  
      "EventType": "STACK_EVENT",  
      "Timestamp": "2025-10-27T17:10:02.923Z",  
      "StartTime": "2025-10-27T17:09:57.537Z",  
      "EndTime": "2025-10-27T17:10:02.923Z"  
    },  
    {
```

```

    "EventId": "2d8c3262-3468-4283-82fb-6e780e9e4f1d",
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/
c3908380-b357-11f0-a97f-0ad08f35df65",
    "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",
    "OperationType": "CREATE_CHANGESET",
    "EventType": "VALIDATION_ERROR",
    "LogicalResourceId": "NotificationBucket",
    "PhysicalResourceId": "",
    "ResourceType": "AWS::S3::Bucket",
    "Timestamp": "2025-10-27T17:10:02.461Z",
    "ValidationFailureMode": "FAIL",
    "ValidationName": "PROPERTY_VALIDATION",
    "ValidationStatus": "FAILED",
    "ValidationStatusReason": "#/NotificationConfiguration/
QueueConfigurations/0: required key [Event] not found",
    "ValidationPath": "/Resources/NotificationBucket/Properties/
NotificationConfiguration/QueueConfigurations/0"
  }
]
}

```

3. Corrigez les erreurs de validation en mettant à jour votre modèle, puis en créant un nouvel ensemble de modifications.
4. Une fois la validation terminée, exécutez l'ensemble de modifications :

```

aws cloudformation execute-change-set \
  --change-set-name MyChangeSet \
  --stack-name MyStack

```

Types de validation

La validation préalable au déploiement inclut les types de contrôles suivants :

- Validation de la syntaxe des propriétés : valide les propriétés des ressources par rapport aux schémas de AWS ressources. Il vérifie les propriétés requises et les valeurs de propriété valides et identifie les combinaisons de propriétés obsolètes ou non prises en charge.
- Détection des conflits de noms de ressources — Vérifie les conflits de dénomination avec les AWS ressources existantes. Il vérifie que les noms de ressources répondent aux exigences de AWS dénomination et identifie les conflits potentiels avant les tentatives de déploiement.

- Validation du vide des compartiments S3 : avertit lorsque vous tentez de supprimer des compartiments S3 contenant des objets. Il fournit le nombre d'objets pour aider à évaluer l'impact de la suppression et à prévenir les échecs de suppression courants dans S3.

Chaque type de validation fournit des messages d'erreur spécifiques et indique l'emplacement des erreurs dans le modèle pour vous aider à résoudre rapidement les problèmes.

Limite de ressources

Les types de ressources suivants ne sont pas pris en charge pour la validation préalable au déploiement :

- `AWS::ApiGatewayV2::ApiGatewayManagedOverrides`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AppMesh::GatewayRoute`
- `AWS::AppMesh::Mesh`
- `AWS::AppMesh::Route`
- `AWS::AppMesh::VirtualGateway`
- `AWS::AppMesh::VirtualNode`
- `AWS::AppMesh::VirtualRouter`
- `AWS::AppMesh::VirtualService`
- `AWS::AppStream::Fleet`
- `AWS::AppStream::Stack`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::AppSync::ApiCache`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AutoScalingPlans::ScalingPlan`
- `AWS::Budgets::Budget`
- `AWS::CertificateManager::Certificate`

- `AWS::Cloud9::EnvironmentEC2`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeBuild::ReportGroup`
- `AWS::CodeBuild::SourceCredential`
- `AWS::CodeCommit::Repository`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodeStar::GitHubRepository`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::OrganizationConfigRule`
- `AWS::Config::RemediationConfiguration`
- `AWS::DAX::Cluster`
- `AWS::DAX::ParameterGroup`
- `AWS::DAX::SubnetGroup`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DLM::LifecyclePolicy`
- `AWS::DMS::Certificate`
- `AWS::DMS::Endpoint`
- `AWS::DMS::EventSubscription`
- `AWS::DMS::ReplicationInstance`
- `AWS::DMS::ReplicationSubnetGroup`

- `AWS::DMS::ReplicationTask`
- `AWS::DocDB::DBCluster`
- `AWS::DocDB::DBClusterParameterGroup`
- `AWS::DocDB::DBInstance`
- `AWS::DocDB::DBSubnetGroup`
- `AWS::DocDB::EventSubscription`
- `AWS::EC2::ClientVpnAuthorizationRule`
- `AWS::EC2::ClientVpnEndpoint`
- `AWS::EC2::ClientVpnRoute`
- `AWS::EC2::ClientVpnTargetNetworkAssociation`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::FSx::FileSystem`
- `AWS::FSx::Snapshot`
- `AWS::FSx::StorageVirtualMachine`
- `AWS::FSx::Volume`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::CustomEntityType`
- `AWS::Glue::DataCatalogEncryptionSettings`

- `AWS::Glue::DataQualityRuleset`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::MLTransform`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::TableOptimizer`
- `AWS::Glue::Workflow`
- `AWS::Greengrass::ConnectorDefinition`
- `AWS::Greengrass::ConnectorDefinitionVersion`
- `AWS::Greengrass::CoreDefinition`
- `AWS::Greengrass::CoreDefinitionVersion`
- `AWS::Greengrass::DeviceDefinition`
- `AWS::Greengrass::DeviceDefinitionVersion`
- `AWS::Greengrass::FunctionDefinition`
- `AWS::Greengrass::FunctionDefinitionVersion`
- `AWS::Greengrass::Group`
- `AWS::Greengrass::GroupVersion`
- `AWS::Greengrass::LoggerDefinition`
- `AWS::Greengrass::LoggerDefinitionVersion`
- `AWS::Greengrass::ResourceDefinition`
- `AWS::Greengrass::ResourceDefinitionVersion`
- `AWS::Greengrass::SubscriptionDefinition`
- `AWS::Greengrass::SubscriptionDefinitionVersion`
- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTThingsGraph::FlowTemplate`

- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisAnalytics::ApplicationReferenceDataSource`
- `AWS::KinesisAnalyticsV2::ApplicationCloudWatchLoggingOption`
- `AWS::KinesisAnalyticsV2::ApplicationOutput`
- `AWS::KinesisAnalyticsV2::ApplicationReferenceDataSource`
- `AWS::LakeFormation::DataLakeSettings`
- `AWS::LakeFormation::Permissions`
- `AWS::LakeFormation::Resource`
- `AWS::ManagedBlockchain::Member`
- `AWS::ManagedBlockchain::Node`
- `AWS::MediaConvert::JobTemplate`
- `AWS::MediaConvert::Preset`
- `AWS::MediaConvert::Queue`
- `AWS::MediaLive::Channel`
- `AWS::MediaLive::Input`
- `AWS::MediaLive::InputSecurityGroup`
- `AWS::MediaStore::Container`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::Pinpoint::ADMChannel`
- `AWS::Pinpoint::APNSChannel`
- `AWS::Pinpoint::APNSSandboxChannel`
- `AWS::Pinpoint::APNSVoipChannel`

- `AWS::Pinpoint::APNSVoipSandboxChannel`
- `AWS::Pinpoint::App`
- `AWS::Pinpoint::ApplicationSettings`
- `AWS::Pinpoint::BaiduChannel`
- `AWS::Pinpoint::Campaign`
- `AWS::Pinpoint::EmailChannel`
- `AWS::Pinpoint::EmailTemplate`
- `AWS::Pinpoint::EventStream`
- `AWS::Pinpoint::GCMChannel`
- `AWS::Pinpoint::PushTemplate`
- `AWS::Pinpoint::Segment`
- `AWS::Pinpoint::SMSChannel`
- `AWS::Pinpoint::SmsTemplate`
- `AWS::Pinpoint::VoiceChannel`
- `AWS::PinpointEmail::ConfigurationSet`
- `AWS::PinpointEmail::ConfigurationSetEventDestination`
- `AWS::PinpointEmail::DedicatedIpPool`
- `AWS::PinpointEmail::Identity`
- `AWS::QLDB::Ledger`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SageMaker::CodeRepository`
- `AWS::SageMaker::EndpointConfig`
- `AWS::SageMaker::Model`
- `AWS::SageMaker::NotebookInstance`

- `AWS::SageMaker::NotebookInstanceLifecycleConfig`
- `AWS::SageMaker::Workteam`
- `AWS::SDB::Domain`
- `AWS::ServiceCatalog::AcceptedPortfolioShare`
- `AWS::ServiceCatalog::LaunchRoleConstraint`
- `AWS::ServiceCatalog::Portfolio`
- `AWS::ServiceCatalog::StackSetConstraint`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptFilter`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::SSM::MaintenanceWindow`
- `AWS::SSM::MaintenanceWindowTarget`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFRegional::ByteMatchSet`
- `AWS::WAFRegional::GeoMatchSet`
- `AWS::WAFRegional::IPSet`
- `AWS::WAFRegional::RateBasedRule`

- `AWS::WAFRegional::RegexPatternSet`
- `AWS::WAFRegional::Rule`
- `AWS::WAFRegional::SizeConstraintSet`
- `AWS::WAFRegional::SqlInjectionMatchSet`
- `AWS::WAFRegional::WebACL`
- `AWS::WAFRegional::WebACLAssociation`
- `AWS::WAFRegional::XssMatchSet`
- `AWS::WorkSpaces::Workspace`
- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::DomainNameAccessAssociation`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::ApplicationEntitlementAssociation`
- `AWS::AppStream::ApplicationFleetAssociation`
- `AWS::AppSync::DomainNameApiAssociation`
- `AWS::AppSync::SourceApiAssociation`
- `AWS::CleanRooms::ConfiguredTableAssociation`
- `AWS::CleanRooms::IdNamespaceAssociation`
- `AWS::CodeGuruReviewer::RepositoryAssociation`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Connect::IntegrationAssociation`
- `AWS::Deadline::QueueFleetAssociation`
- `AWS::Deadline::QueueLimitAssociation`
- `AWS::EC2::EIPAssociation`
- `AWS::EC2::EnclaveCertificateIamRoleAssociation`
- `AWS::EC2::GatewayRouteTableAssociation`
- `AWS::EC2::IPAMResourceDiscoveryAssociation`

- `AWS::EC2::IpPoolRouteTableAssociation`
- `AWS::EC2::LocalGatewayRouteTableVPCAssociation`
- `AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation`
- `AWS::EC2::NetworkInterfaceAttachment`
- `AWS::EC2::RouteServerAssociation`
- `AWS::EC2::SecurityGroupVpcAssociation`
- `AWS::EC2::SubnetNetworkAclAssociation`
- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::TransitGatewayMulticastDomainAssociation`
- `AWS::EC2::TransitGatewayPeeringAttachment`
- `AWS::EC2::TransitGatewayRouteTableAssociation`
- `AWS::EC2::TransitGatewayVpcAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VolumeAttachment`
- `AWS::ECS::ClusterCapacityProviderAssociations`
- `AWS::EKS::PodIdentityAssociation`
- `AWS::FSx::DataRepositoryAssociation`
- `AWS::FSx::S3AccessPointAttachment`
- `AWS::GlobalAccelerator::CrossAccountAttachment`
- `AWS::LakeFormation::TagAssociation`
- `AWS::NetworkFirewall::VpcEndpointAssociation`
- `AWS::NetworkManager::ConnectAttachment`
- `AWS::NetworkManager::CustomerGatewayAssociation`
- `AWS::NetworkManager::DirectConnectGatewayAttachment`
- `AWS::NetworkManager::LinkAssociation`
- `AWS::NetworkManager::SiteToSiteVpnAttachment`
- `AWS::NetworkManager::TransitGatewayRouteTableAttachment`

- `AWS::NetworkManager::VpcAttachment`
- `AWS::Notifications::ChannelAssociation`
- `AWS::Notifications::ManagedNotificationAccountContactAssociation`
- `AWS::Notifications::ManagedNotificationAdditionalChannelAssociation`
- `AWS::Notifications::OrganizationalUnitAssociation`
- `AWS::ResourceExplorer2::DefaultViewAssociation`
- `AWS::Route53Profiles::ProfileAssociation`
- `AWS::Route53Profiles::ProfileResourceAssociation`
- `AWS::Route53Resolver::FirewallRuleGroupAssociation`
- `AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation`
- `AWS::Route53Resolver::ResolverRuleAssociation`
- `AWS::SSM::Association`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecurityHub::PolicyAssociation`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::ServiceActionAssociation`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkServiceAssociation`
- `AWS::VpcLattice::ServiceNetworkVpcAssociation`
- `AWS::WAFv2::WebACLAssociation`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::WorkspacesInstances::VolumeAssociation`
- `AWS::IAM::Policy`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::QueuePolicy`

- `AWS::EC2::NetworkAclEntry`
- `AWS::EC2::VPNGatewayRoutePropagation`
- `AWS::CloudFormation::Stack`
- `AWS::CloudWatch::MetricStream`
- `AWS::WorkSpaces::ConnectionAlias`
- `AWS::IoT::ProvisioningTemplate`
- `AWS::MediaPackage::Channel`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::Route53Resolver::ResolverQueryLoggingConfig`
- `AWS::NetworkManager::TransitGatewayRegistration`
- `AWS::ImageBuilder::Image`
- `AWS::Config::ConformancePack`
- `AWS::S3::AccessPoint`
- `AWS::CodeStarConnections::Connection`
- `AWS::CloudFront::CachePolicy`
- `AWS::FMS::NotificationChannel`
- `AWS::ImageBuilder::InfrastructureConfiguration`
- `AWS::Detective::Graph`
- `AWS::EC2::CarrierGateway`
- `AWS::CloudWatch::CompositeAlarm`
- `AWS::CodeArtifact::Repository`
- `AWS::GroundStation::DataflowEndpointGroup`
- `AWS::ElasticLoadBalancingV2::Listener`
- `AWS::ImageBuilder::ImageRecipe`
- `AWS::NetworkManager::Device`
- `AWS::Kendra::DataSource`
- `AWS::Timestream::Database`
- `AWS::CodeGuruProfiler::ProfilingGroup`
- `AWS::Lambda::EventSourceMapping`

- `AWS::ECR::Repository`
- `AWS::WAFv2::IPSet`
- `AWS::GameLift::Alias`
- `AWS::IoTSiteWise::Asset`
- `AWS::OpsWorksCM::Server`
- `AWS::IoT::Authorizer`
- `AWS::WAFv2::RuleGroup`
- `AWS::NetworkManager::Site`
- `AWS::ResourceGroups::Group`
- `AWS::MediaPackage::PackagingConfiguration`
- `AWS::ImageBuilder::ImagePipeline`
- `AWS::ECS::TaskDefinition`
- `AWS::Macie::CustomDataIdentifier`
- `AWS::MediaPackage::OriginEndpoint`
- `AWS::Logs::LogGroup`
- `AWS::CodeArtifact::Domain`
- `AWS::Kendra::Faq`
- `AWS::ECS::TaskSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::ECS::Cluster`
- `AWS::SSO::Assignment`
- `AWS::GlobalAccelerator::Listener`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::RDS::DBProxy`
- `AWS::EC2::FlowLog`
- `AWS::ImageBuilder::Component`
- `AWS::CloudFront::RealtimeLogConfig`
- `AWS::NetworkManager::GlobalNetwork`
- `AWS::RDS::DBProxyTargetGroup`

- `AWS::WAFv2::WebACL`
- `AWS::IVS::StreamKey`
- `AWS::IVS::PlaybackKeyPair`
- `AWS::Macie::Session`
- `AWS::Route53::HealthCheck`
- `AWS::Synthetics::Canary`
- `AWS::Lambda::CodeSigningConfig`
- `AWS::EFS::AccessPoint`
- `AWS::Timestream::Table`
- `AWS::MediaPackage::PackagingGroup`
- `AWS::ECS::PrimaryTaskSet`
- `AWS::Config::ConfigurationAggregator`
- `AWS::GroundStation::Config`
- `AWS::IoTSiteWise::AssetModel`
- `AWS::SES::ConfigurationSet`
- `AWS::ImageBuilder::DistributionConfiguration`
- `AWS::Config::OrganizationConformancePack`
- `AWS::EC2::LocalGatewayRoute`
- `AWS::KMS::Key`
- `AWS::Detective::MemberInvitation`
- `AWS::EKS::FargateProfile`
- `AWS::MediaPackage::Asset`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::Macie::FindingsFilter`
- `AWS::IoT::Certificate`
- `AWS::SageMaker::MonitoringSchedule`
- `AWS::IVS::Channel`
- `AWS::Kendra::Index`
- `AWS::EventSchemas::RegistryPolicy`

- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::GlobalAccelerator::Accelerator`
- `AWS::EC2::PrefixList`
- `AWS::GameLift::GameServerGroup`
- `AWS::NetworkManager::Link`
- `AWS::EFS::FileSystem`
- `AWS::Route53::HostedZone`
- `AWS::GroundStation::MissionProfile`
- `AWS::KMS::Alias`
- `AWS::FMS::Policy`
- `AWS::SSO::PermissionSet`
- `AWS::StepFunctions::StateMachine`
- `AWS::QLDB::Stream`
- `AWS::IoTSiteWise::Gateway`
- `AWS::ECS::Service`
- `AWS::ECS::CapacityProvider`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::EC2Fleet`
- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Topic`
- `AWS::SNS::Subscription`
- `AWS::RDS::DBInstance`

- `AWS::RDS::DBParameterGroup`
- `AWS::RDS::DBCluster`
- `AWS::RDS::DBClusterParameterGroup`
- `AWS::RDS::DBSubnetGroup`
- `AWS::RDS::EventSubscription`
- `AWS::RDS::GlobalCluster`
- `AWS::RDS::OptionGroup`
- `AWS::Neptune::DBInstance`
- `AWS::Neptune::DBParameterGroup`
- `AWS::Neptune::DBCluster`
- `AWS::Neptune::DBClusterParameterGroup`
- `AWS::Neptune::DBSubnetGroup`
- `AWS::Redshift::Cluster`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::Redshift::ClusterSubnetGroup`
- `AWS::Redshift::EndpointAccess`
- `AWS::Redshift::EndpointAuthorization`
- `AWS::Redshift::EventSubscription`
- `AWS::Redshift::ScheduledAction`
- `AWS::ElasticCache::SubnetGroup`

Mettre à jour directement les piles

Si vous souhaitez déployer rapidement des mises à jour dans une pile, procédez à une mise à jour directe. Avec une mise à jour directe, vous soumettez un modèle ou des paramètres d'entrée qui spécifient les mises à jour des ressources de la pile, et vous les déployez CloudFormation immédiatement. Si vous voulez utiliser un modèle pour effectuer vos mises à jour, vous pouvez modifier le modèle actuel et le stocker localement ou dans un compartiment Amazon S3.

Pour les propriétés de ressource qui ne prennent en charge les mises à jour, vous devez conserver les valeurs actuelles. Pour prévisualiser les modifications que CloudFormation seront apportées à

avant de la mettre à jour, utilisez des ensembles de modifications. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

Lors de la mise à jour d'une pile, les ressources CloudFormation peuvent être interrompues ou remplacées par des ressources mises à jour, selon les propriétés que vous mettez à jour. Pour plus d'informations sur les mises à jour de ressources, consultez [Comprendre les comportements de mise à jour des ressources d'une pile](#).

Pour mettre à jour une pile (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sur la page Piles, sélectionnez la pile en cours d'exécution que vous voulez mettre à jour.
4. Dans le volet des détails de la pile, choisissez Update (Mettre à jour).
5. Si vous n'avez pas modifié le modèle de pile, sélectionnez Utiliser le modèle existant, puis choisissez Suivant.

Si vous avez modifié le modèle, sélectionnez Remplacer le modèle existant et indiquez l'emplacement du modèle mis à jour dans la section Spécifier le modèle:

- Pour un modèle stocké localement sur votre ordinateur, sélectionnez Upload a template file (Charger un fichier de modèle). Choisissez Choose File (Choisir un fichier) pour accéder au fichier et le sélectionner, puis choisissez Suivant.

Note

Si vous chargez un fichier modèle local, CloudFormation chargez-le dans un compartiment Amazon Simple Storage Service (Amazon S3) de votre Compte AWS. Si vous ne possédez pas encore de compartiment S3 créé par CloudFormation, il crée un compartiment unique pour chaque région dans laquelle vous téléchargez un fichier modèle. Si vous possédez déjà un compartiment S3 créé par CloudFormation in your Compte AWS, CloudFormation ajoute le modèle à ce compartiment.

Considérations à prendre en compte concernant les compartiments S3 créés par CloudFormation

- Les compartiments sont accessibles à toute personne disposant des autorisations Amazon S3 dans votre Compte AWS.

- CloudFormation crée les compartiments avec le chiffrement côté serveur activé par défaut, chiffrant ainsi tous les objets stockés dans le compartiment.

Vous pouvez gérer directement les options de chiffrement pour les CloudFormation compartiments créés ; par exemple, en utilisant la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>, ou le AWS CLI. Pour plus d'informations, consultez [Configuration du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

- Pour utiliser votre propre compartiment et gérer ses autorisations, vous pouvez importer manuellement les modèles dans Amazon S3. Lorsque que vous créez ou mettez à jour une pile, spécifiez l'URL Amazon S3 d'un fichier de modèle.

- Pour un modèle stocké dans un compartiment Amazon S3, choisissez URL Amazon S3. Entrez ou collez l'URL du modèle, puis choisissez Suivant.

Si votre modèle se trouve dans un compartiment dans lequel la gestion des versions est activée, vous pouvez spécifier une version particulière du modèle en ajoutant ?versionId=*version-id* à l'URL. Pour plus d'informations, consultez [Travail avec des objets dans un compartiment avec gestion des versions activée](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

Si des problèmes de syntaxe sont détectés, la console affiche des messages d'erreur qui vous aident à corriger le modèle.

6. Si votre modèle contient des paramètres, sur la page Specify stack details (Spécifier les détails de pile), vous pouvez entrer ou modifier les valeurs des paramètres, puis choisissez Suivant.

CloudFormation remplit chaque paramètre avec la valeur actuellement définie dans la pile, à l'exception des paramètres déclarés avec l'NoEchoattribut ; toutefois, vous pouvez toujours utiliser les valeurs actuelles en cochant la case Utiliser la valeur existante.

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles, en plus de l'utilisation de paramètres dynamiques pour gérer les secrets, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

7. Sur la page Configurer les options de pile, vous pouvez mettre à jour les balises et les autorisations appliquées à la pile et modifier les options avancées telles que la politique de pile

ou la configuration de restauration, ou mettre à jour la rubrique de notification Amazon SNS. Pour plus d'informations sur ces options, consultez [Configuration des options de pile](#).

8. Si votre modèle contient des ressources IAM, dans la section Capacités, choisissez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous autorisez l'utilisation de ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
9. Choisissez Next (Suivant) pour continuer.
10. Passez en revue les informations de la pile et les modifications que vous avez soumises.

Vérifiez que vous avez soumis les informations correctes, telles que l'URL du modèle ou les valeurs de paramètres appropriées.

Dans la section Aperçu du jeu de modifications, vérifiez qu'il CloudFormation apportera toutes les modifications attendues. Par exemple, vous pouvez vérifier que CloudFormation les ressources que vous aviez l'intention d'ajouter, de supprimer ou de modifier sont ajoutées, supprimées et modifiées. CloudFormation génère cet aperçu en créant un ensemble de modifications pour la pile. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

11. Lorsque vous êtes satisfait de vos modifications, choisissez Update stack (Mettre à jour la pile).

 Note

À ce stade, vous avez aussi la possibilité d'afficher le jeu de modifications pour vérifier de façon plus complète vos mises à jour proposées. Pour ce faire, choisissez Afficher l'ensemble de modifications au lieu de Mettre à jour la pile. CloudFormation affiche l'ensemble de modifications généré en fonction de vos mises à jour. Lorsque vous êtes prêt à effectuer la mise à jour de la pile, choisissez Exécuter.

CloudFormation affiche la page des détails de la pile correspondant à votre pile, avec le volet Événements sélectionné. Votre pile a désormais le statut UPDATE_IN_PROGRESS. Une CloudFormation fois la mise à jour de la pile terminée avec succès, l'état de la pile est défini sur UPDATE_COMPLETE.

Si la mise à jour de la pile échoue, CloudFormation ; annule automatiquement les modifications et définit le statut de la pile sur UPDATE_ROLLBACK_COMPLETE.

 Note

Vous pouvez annuler une mise à jour tant qu'elle est dans l'état `UPDATE_IN_PROGRESS`. Pour de plus amples informations, veuillez consulter [Annuler la mise à jour d'une pile](#).

Pour mettre à jour une pile à l'aide de la ligne de commande

Vous pouvez utiliser l'une des commandes suivantes :

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

Pour des exemples d'utilisation de la ligne de commande pour mettre à jour une pile, consultez [Exemples de commandes d'opération de CloudFormation pile pour AWS CLI et PowerShell](#).

Annuler la mise à jour d'une pile

Une fois qu'une mise à jour de la pile a commencé, vous pouvez l'annuler tant que la pile indique encore l'état `UPDATE_IN_PROGRESS`. Une fois qu'une mise à jour est terminée, vous ne pouvez plus l'annuler. Vous pouvez, toutefois, mettre à jour une pile à nouveau afin de rétablir les paramètres précédents.

Si vous annulez la mise à jour d'une pile, c'est la configuration qui existait juste avant le lancement de la mise à jour qui est restaurée.

Rubriques

- [Pour annuler une mise à jour de pile \(console\)](#)
- [Pour annuler une mise à jour de pile \(AWS CLI\)](#)

Pour annuler une mise à jour de pile (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de la pile.

3. Sur la page Piles, sélectionnez la pile en cours de mise à jour. Son statut doit être `UPDATE_IN_PROGRESS`.
4. Choisissez Stack actions (Actions de pile), puis Annuler la mise à jour de la pile.
5. Pour continuer à annuler la mise à jour, choisissez Annuler la mise à jour. Sinon, choisissez Annuler pour reprendre la mise à jour.

La pile passe à l'état `UPDATE_ROLLBACK_IN_PROGRESS`. Une fois l'annulation de la mise à jour terminée, la pile est définie sur `UPDATE_ROLLBACK_COMPLETE`.

Pour annuler une mise à jour de pile (AWS CLI)

Utilisez la commande [cancel-update-stack](#) pour annuler une mise à jour. Pour de plus amples informations, veuillez consulter [Annuler la mise à jour d'une pile](#).

Supprimer une pile de la CloudFormation console

Si vous n'avez plus besoin des ressources d'une pile, vous pouvez supprimer toute la pile.

Lorsque vous supprimez une pile, toutes les ressources CloudFormation de cette pile sont supprimées, sauf si vous avez utilisé une politique de suppression pour conserver des ressources spécifiques. Pour plus d'informations, voir la section consacrée à l'[Attribut DeletionPolicy](#).

Suppression d'une pile (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de la pile.
3. Sur la page Piles, choisissez la pile que vous souhaitez supprimer. La pile doit être en cours d'exécution.
4. Sélectionnez Supprimer.
5. Lorsque vous êtes invité à confirmer l'opération, choisissez Supprimer.

Note

L'opération de suppression de pile ne peut pas être arrêtée une fois qu'elle a commencé. La pile passe à l'état `DELETE_IN_PROGRESS`.

Une fois que la suppression est terminée, la pile indique l'état `DELETE_COMPLETE`. Les piles dans `DELETE_COMPLETE` cet état ne sont pas affichées dans la CloudFormation console par défaut. Pour afficher les piles supprimées, vous devez modifier le filtre d'affichage des piles, comme décrit dans [Afficher les piles supprimées depuis la console CloudFormation](#).

Suppression forcée d'une pile (console)

La suppression d'une pile peut échouer lorsqu'une ressource de la pile ne peut pas être supprimée. Par exemple, la suppression d'une ressource dont dépend également une autre pile CloudFormation échouera. Toutes les ressources qui n'ont pas été supprimées resteront tant que vous ne réussirez pas à supprimer la pile. Si la suppression échoue et renvoie un état `DELETE_FAILED`, vous pouvez réessayer la suppression selon deux méthodes.

1. Sur la page Stacks de la CloudFormation console, choisissez la pile dont vous souhaitez forcer la suppression.
2. Dans le volet des détails de la pile, choisissez Réessayer la suppression.
3. Choisissez l'une des options suivantes :
 - Supprimer cette pile tout en retenant les ressources : cette option vous permet de sélectionner les ressources qui avaient échoué à la suppression et que vous souhaitez retenir lors de la suppression forcée de la pile.
 - Forcer la suppression de toute cette pile : cette option retient toutes les ressources qui n'ont pas pu être supprimées et retient également les dépendances de ces ressources.
4. Choisissez Supprimer pour lancer le processus de suppression forcée avec vos sélections.

Consultation des ressources retenues (console)

Après avoir supprimé la pile, vous pouvez consulter les ressources retenues dans la console.

1. Dans la liste des piles, choisissez Filtrer par état, puis sélectionnez Supprimé.
2. Sélectionnez la pile supprimée.
3. Sélectionnez l'onglet Ressources.
4. Toutes les ressources retenues afficheront l'État `DELETE_SKIPPED`.
5. Choisissez la ressource retenue que vous souhaitez consulter.

Suppression d'une pile à l'aide de la ligne de commande

Vous pouvez utiliser l'une des commandes suivantes :

- [delete-stack](#) (AWS CLI)
- [Remove-CFNStack](#) (AWS Tools for Windows PowerShell)

Pour des exemples de suppression de pile à l'aide de la ligne de commande, consultez [Exemples de commandes d'opération de CloudFormation pile pour AWS CLI et PowerShell](#).

Ressources connexes

Pour obtenir de l'aide pour résoudre les problèmes de suppression de piles, consultez la rubrique relative à la résolution des problèmes [Échec de la suppression de la pile](#).

Pour obtenir des informations sur la protection des piles contre les suppressions accidentelles, consultez [CloudFormation Empêcher les piles d'être supprimées](#).

Afficher les piles supprimées depuis la console CloudFormation

Par défaut, la CloudFormation console n'affiche pas les piles dont le statut est `DELETE_COMPLETE`. Pour afficher des informations sur les piles supprimées, vous devez modifier le paramètre d'affichage des piles.

Pour afficher les piles supprimées

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de la pile supprimée.
3. Sur la page Pile, choisissez Supprimé dans le menu déroulant Filtrer par état.

CloudFormation répertorie toutes vos piles dont le statut est de. `DELETE_COMPLETE`

Consultez aussi

- [Supprimer une pile de la CloudFormation console](#)
- [Afficher les informations relatives à la pile depuis la CloudFormation console](#)

Surveiller la progression de la pile

Cette section décrit comment surveiller un déploiement de stack en cours. CloudFormation fournit une liste chronologique détaillée des événements de déploiement, indiquant la progression et les éventuels problèmes rencontrés au cours du déploiement.

Rubriques

- [Afficher les événements CloudFormation liés à la pile](#)
- [Afficher les événements de la pile par opération](#)
- [Afficher la chronologie d'un déploiement de CloudFormation stack](#)
- [Comprendre les événements de création de CloudFormation piles](#)
- [Surveiller la progression d'une mise à jour de pile](#)
- [Poursuite de la restauration d'une mise à jour](#)
- [Détermination de la cause d'un échec de pile](#)
- [Choisissez comment gérer les défaillances lors de l'approvisionnement des ressources](#)

Afficher les événements CloudFormation liés à la pile

Vous pouvez afficher les événements de pile pour surveiller la progression et l'état de votre pile et des ressources qu'elle contient. Les événements de pile vous aident à comprendre quand les ressources sont créées, mises à jour ou supprimées, et si le déploiement de la pile se déroule comme prévu.

Rubriques

- [Afficher les événements de pile \(console\)](#)
- [Afficher les événements de la pile \(AWS CLI\)](#)
- [Codes d'état de la pile](#)

Afficher les événements de pile (console)

Pour afficher les événements de pile

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation située en haut de l'écran, choisissez la Région AWS dans laquelle vous avez créé la pile.

3. Sur la page Stacks de la CloudFormation console, sélectionnez le nom de la pile. CloudFormation affiche les détails de la pile sélectionnée.
4. Choisissez l'onglet Événements pour afficher les événements CloudFormation de pile générés pour votre pile.

CloudFormation actualise automatiquement les événements de la pile toutes les minutes. CloudFormation Affiche également le badge Nouveaux événements disponibles lorsque de nouveaux événements de pile se produisent. Sélectionnez l'icône d'actualisation pour charger ces événements dans la liste. En regardant les événements de création de la pile, vous pouvez comprendre la séquence d'événements qui conduit à sa création (ou à son échec, si vous êtes en phase de débogage).

Lorsque la pile est en cours de création, elle est répertoriée sur la page Piles avec le statut `CREATE_IN_PROGRESS`. Une fois que la pile a été créée avec succès, son statut passe à `CREATE_COMPLETE`.

Pour plus d'informations, consultez [Comprendre les événements de création de CloudFormation piles](#) et [Surveiller la progression d'une mise à jour de pile](#).

Afficher les événements de la pile (AWS CLI)

Vous pouvez également utiliser la [describe-stack-events](#) commande lors de la création de la pile pour afficher les événements tels qu'ils sont signalés.

La commande `describe-stack-events` suivante décrit les événements de la pile *my-stack*.

```
aws cloudformation describe-stack-events --stack-name my-stack
```

Voici un exemple de réponse.

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "7b755820-7edf-11f0-ab15-0673b09f3847",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
```

```

    "ResourceType": "AWS::CloudFormation::Stack",
    "Timestamp": "2025-08-21T22:37:56.243000+00:00",
    "ResourceStatus": "CREATE_COMPLETE",
    "ClientRequestToken": "token"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
    "EventId": "WebServer-CREATE_COMPLETE-2025-08-21T22:37:54.356Z",
    "StackName": "my-stack",
    "LogicalResourceId": "WebServer",
    "PhysicalResourceId": "i-099df76cb31b866a9",
    "ResourceType": "AWS::EC2::Instance",
    "Timestamp": "2025-08-21T22:37:54.356000+00:00",
    "ResourceStatus": "CREATE_COMPLETE",
    "ResourceProperties": "{\"UserData\":
\\\"IyEvYm1uL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYm1uL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[{\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"}]}\",
    \"ClientRequestToken\": \"token\"
  },
  {
    \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
    \"EventId\": \"WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:31.226Z\",
    \"StackName\": \"my-stack\",
    \"LogicalResourceId\": \"WebServer\",
    \"PhysicalResourceId\": \"i-099df76cb31b866a9\",
    \"ResourceType\": \"AWS::EC2::Instance\",
    \"Timestamp\": \"2025-08-21T22:37:31.226000+00:00\",
    \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
    \"ResourceStatusReason\": \"Resource creation Initiated\",
    \"ResourceProperties\": \"{\\\"UserData\\\":
\\\"IyEvYm1uL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYm1uL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[{\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"}]}\",
    \"ClientRequestToken\": \"token\"
  },
  {
    \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
    \"EventId\": \"WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:29.210Z\",

```

```

      "StackName": "my-stack",
      "LogicalResourceId": "WebServer",
      "PhysicalResourceId": "",
      "ResourceType": "AWS::EC2::Instance",
      "Timestamp": "2025-08-21T22:37:29.210000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceProperties": "{\"UserData\": \"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I\", \"ImageId\": \"ami-0bbc328167dee8f3c\", \"InstanceType\": \"t2.micro\", \"SecurityGroupIds\": [\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"], \"Tags\": [{\"Value\": \"Bootstrap Tutorial Web Server\", \"Key\": \"Name\"}]}\", \"ClientRequestToken\": \"token\"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServerSecurityGroup-CREATE_COMPLETE-2025-08-21T22:37:28.803Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "my-stack-WebServerSecurityGroup-n8A43bQT1ty2",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:28.803000+00:00",
      "ResourceStatus": "CREATE_COMPLETE",
      "ResourceProperties": "{\"GroupDescription\": \"Allow HTTP access from my IP address\", \"SecurityGroupIngress\": [{\"CidrIp\": \"0.0.0.0/0\", \"Description\": \"HTTP\", \"FromPort\": \"80\", \"ToPort\": \"80\", \"IpProtocol\": \"tcp\"}]}\", \"ClientRequestToken\": \"token\"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServerSecurityGroup-CREATE_IN_PROGRESS-2025-08-21T22:37:22.626Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "my-stack-WebServerSecurityGroup-n8A43bQT1ty2",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:22.626000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceStatusReason": "Resource creation Initiated",
      "ResourceProperties": "{\"GroupDescription\": \"Allow HTTP access from my IP address\", \"SecurityGroupIngress\": [{\"CidrIp\": \"0.0.0.0/0\", \"Description\": \"HTTP\", \"FromPort\": \"80\", \"ToPort\": \"80\", \"IpProtocol\": \"tcp\"}]}\",

```

```
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:20.186Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:20.186000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceProperties": "{\"GroupDescription\": \"Allow HTTP access from my IP
address\", \"SecurityGroupIngress\": [{\"CidrIp\": \"0.0.0.0/0\", \"Description\": \"HTTP
\", \"FromPort\": \"80\", \"ToPort\": \"80\", \"IpProtocol\": \"tcp\"}]}",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "64740fe0-7edf-11f0-8a36-06453a64f325",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-
region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2025-08-21T22:37:17.819000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceStatusReason": "User Initiated",
      "ClientRequestToken": "token"
    }
  ]
}
```

Les événements les plus récents apparaissent en premier. Le tableau suivant décrit les champs renvoyés par la commande `describe-stack-events` :

Champ	Description
EventId	Identifiant de l'événement.

Champ	Description
StackName	Nom de la pile à laquelle l'événement correspond.
StackId	Identifiant de la pile à laquelle l'événement correspond.
LogicalResourceId	Identifiant logique de la ressource.
PhysicalResourceId	Identifiant physique de la ressource.
ResourceProperties	Propriétés de la ressource.
ResourceType	Type de ressource.
Timestamp	Moment où l'événement est survenu.
ResourceStatus	Le statut de la ressource, qui peut être l'un des codes d'état suivants : CREATE_COMPLETE CREATE_FAILED CREATE_IN_PROGRESS DELETE_COMPLETE DELETE_FAILED DELETE_IN_PROGRESS DELETE_SKIPPED IMPORT_COMPLETE IMPORT_IN_PROGRESS IMPORT_ROLLBACK_COMPLETE IMPORT_ROLLBACK_FAILED IMPORT_ROLLBACK_IN_PROGRESS REVIEW_IN_PROGRESS ROLLBACK_COMPLETE ROLLBACK_FAILED ROLLBACK_IN_PROGRESS UPDATE_COMPLETE UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_FAILED UPDATE_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_FAILED UPDATE_ROLLBACK_IN_PROGRESS L'état DELETE_SKIPPED s'applique aux ressources dont l'attribut de la politique de suppression correspond à « retainer ».
DetailedStatus	État détaillé de la pile. Si CONFIGURATION_COMPLETE est présent, la phase de configuration des ressources de la pile est terminée et la stabilisation des ressources est en cours.
ResourceStatusReason	Plus d'informations sur cet état.

Codes d'état de la pile

Le tableau suivant décrit les codes d'état d'une pile :

État de la pile et statut détaillé facultatif	Description
CREATE_COMPLETE	Réussite de la création d'une ou de plusieurs piles.
CREATE_IN_PROGRESS	Création en cours d'une ou de plusieurs piles.
CREATE_FAILED	Échec de la création d'une ou de plusieurs piles. Affichez les événements de la pile pour voir les messages d'erreur associés. Les raisons possibles d'un échec de création incluent des autorisations insuffisantes pour utiliser toutes les ressources de la pile, des valeurs de paramètres rejetées par un AWS service ou un délai d'attente lors de la création des ressources.
DELETE_COMPLETE	Réussite de la suppression d'une ou de plusieurs piles. Les piles supprimées sont conservées et consultables pendant 90 jours.
DELETE_FAILED	Échec de la suppression d'une ou de plusieurs piles. Comme la suppression a échoué, certaines ressources sont peut-être encore en cours d'exécution. Cependant, vous ne pouvez pas utiliser ni mettre à jour la pile. Supprimez la pile à nouveau ou consultez les événements correspondants pour voir les messages d'erreur associés.
DELETE_IN_PROGRESS	Suppression en cours d'une ou de plusieurs piles.
REVIEW_IN_PROGRESS	Création continue d'une ou de plusieurs piles avec un StackId attendu, mais sans aucun modèle ni ressource.

État de la pile et statut détaillé facultatif	Description
	 Important Une pile associée à ce code de statut est comptabilisée dans le nombre maximal possible de piles.
ROLLBACK_COMPLETE	Réussite de la suppression d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile. La pile revient à son état de fonctionnement précédent. Toutes les ressources qui ont été créées au cours de l'action de création de la pile sont supprimées. Ce statut existe uniquement après un échec de création de la pile. Il signifie que toutes les opérations de la pile partiellement créée ont été correctement nettoyées. Dans ce cas-là, seule une opération de suppression peut être effectuée.
ROLLBACK_FAILED	Échec de la suppression d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile. Supprimez la pile ou consultez les événements correspondants pour voir les messages d'erreur associés.
ROLLBACK_IN_PROGRESS	Suppression en cours d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile.
UPDATE_COMPLETE	Réussite de la mise à jour d'une ou de plusieurs piles.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Suppression en cours d'anciennes ressources pour une ou plusieurs piles après une mise à jour réussie. Pour les mises à jour de pile qui nécessitent le remplacement de ressources, CloudFormation crée d'abord les nouvelles ressources, puis supprime les anciennes ressources afin de réduire les interruptions de votre pile. Dans cet état, la pile a été mise à jour et est utilisable, mais les anciennes ressources CloudFormation sont toujours supprimées.

État de la pile et statut détaillé facultatif	Description
UPDATE_FAILED	Réussite de la mise à jour d'une ou de plusieurs piles. Affichez les événements de la pile pour voir les messages d'erreur associés.
UPDATE_IN_PROGRESS	Mise à jour en cours d'une ou de plusieurs piles.
UPDATE_ROLLBACK_COMPLETE	Réussite de la restauration de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Suppression en cours des nouvelles ressources d'une ou de plusieurs piles après un échec de mise à jour. Dans cet état, la pile a été rétablie à son état de fonctionnement précédent et est utilisable, mais CloudFormation les nouvelles ressources créées lors de la mise à jour de la pile sont toujours supprimées.
UPDATE_ROLLBACK_FAILED	Échec de la restauration de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour. Dans ce cas là, vous pouvez supprimer la pile ou poursuivre la restauration . Vous devrez peut-être corriger les erreurs avant que votre pile puisse revenir à un état de fonctionnement. Vous pouvez également contacter Support pour restaurer la pile dans un état utilisable.
UPDATE_ROLLBACK_IN_PROGRESS	Restauration en cours de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour.
IMPORT_IN_PROGRESS	L'opération d'importation est en cours.
IMPORT_COMPLETE	L'opération d'importation a été exécutée avec succès pour toutes les ressources de la pile qui prennent en charge <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	L'importation sera restaurée à la configuration précédente du modèle.

État de la pile et statut détaillé facultatif	Description
IMPORT_ROLLBACK_FAILED	L'opération de restauration de l'importation a échoué pour au moins une ressource de la pile. Les résultats seront disponibles pour les ressources importées CloudFormation avec succès.
IMPORT_ROLLBACK_COMPLETE	L'importation a été restaurée avec succès à la configuration du modèle précédente.

Afficher les événements de la pile par opération

Vous pouvez consulter les événements de la pile regroupés par opération afin de mieux comprendre la séquence et l'étendue des modifications apportées à votre pile. Le regroupement basé sur les opérations vous permet de suivre ensemble les événements connexes, ce qui facilite le suivi des progrès et la résolution des problèmes lorsqu'ils surviennent.

Chaque opération de pile (création, mise à jour, suppression, annulation) se voit attribuer un identifiant d'opération unique qui regroupe tous les événements connexes. Cela vous permet de vous concentrer sur des opérations spécifiques et d'identifier rapidement la cause première des défaillances.

Rubriques

- [Conditions préalables](#)
- [Afficher les événements de la pile par opération \(console\)](#)
- [Afficher les événements de la pile par opération \(AWS CLI\)](#)
- [Codes d'état de la pile](#)

Conditions préalables

Pour utiliser `DescribeEvents` l'API, vous devez disposer des autorisations IAM nécessaires pour `:DescribeEvents`.

Afficher les événements de la pile par opération (console)

Pour afficher les événements de la pile regroupés par opération

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l'option dans Région AWS laquelle vous avez créé la pile.
3. Sur la page Stacks, sélectionnez le nom de la pile. CloudFormation affiche les détails de la pile sélectionnée.
4. Choisissez l'onglet Événements pour afficher les événements CloudFormation de pile générés pour votre pile.
5. Les événements sont automatiquement regroupés par ID d'opération. Chaque opération apparaît sous la forme d'une section extensible indiquant le type, le statut et l'horodatage de l'opération.
6. Cliquez sur un ID d'opération pour ouvrir une vue détaillée affichant uniquement les événements liés à cette opération spécifique.
7. Dans la vue détaillée des opérations, cochez la case Afficher les échecs uniquement pour afficher uniquement les événements ayant échoué à des fins d'analyse des causes premières.

CloudFormation actualise automatiquement les événements de la pile toutes les minutes. Le badge Nouveaux événements disponibles apparaît lorsque de nouveaux événements de pile se produisent. Sélectionnez l'icône d'actualisation pour charger ces événements dans la liste.

En visualisant les événements de la pile regroupés par opération, vous pouvez comprendre la séquence des événements pour chaque opération et identifier rapidement l'opération spécifique à l'origine des problèmes (si vous êtes en train de déboguer votre pile).

Lorsque votre opération de stack est en cours d'exécution, elle est répertoriée avec un statut de `CREATE_IN_PROGRESS`, `UPDATE_IN_PROGRESS`, ou `DELETE_IN_PROGRESS`. Une fois votre opération terminée avec succès, son statut passe à `CREATE_COMPLETE`, `UPDATE_COMPLETE`, ou `DELETE_COMPLETE`.

Pour plus d'informations, consultez [Comprendre les événements de création de CloudFormation piles](#) et [Surveiller la progression d'une mise à jour de pile](#).

Afficher les événements de la pile par opération (AWS CLI)

Vous pouvez utiliser la `describe-events` commande avec le filtrage des identifiants d'opération pour afficher les événements relatifs à des opérations spécifiques.

Obtenir la dernière opération IDs

La description de la pile disponible via l'API `describe-stacks` inclut désormais des `LastOperations` informations indiquant les opérations récentes IDs et leurs types. Cela vous permet d'identifier rapidement les opérations qui ont eu lieu et leur état actuel sans avoir à parcourir les journaux d'événements.

```
aws cloudformation describe-stacks --stack-name MyStack
```

Voici un exemple de réponse montrant que la dernière opération était une annulation suite à l'échec d'une opération de mise à jour.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/07580010-bb79-11f0-8f6c-0289bb5c804f",
      "StackName": "MyStack",
      "Description": "A simple CloudFormation template to create an S3 bucket.",
      "CreationTime": "2025-11-07T01:28:13.778000+00:00",
      "LastUpdatedTime": "2025-11-07T01:43:39.838000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "UPDATE_ROLLBACK_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Tags": [],
      "EnableTerminationProtection": false,
      "DriftInformation": {
        "StackDriftStatus": "NOT_CHECKED"
      },
      "LastOperations": [
        {
          "OperationType": "ROLLBACK",
          "OperationId": "d0f12313-7bdb-414d-a879-828a99b36f29"
        },
        {
```

```

        "OperationType": "UPDATE_STACK",
        "OperationId": "1c211b5a-4538-4dc9-bfed-e07734371e57"
      }
    ]
  }
}

```

Filtrer les événements par ID d'opération

La `describe-events` commande suivante décrit les événements relatifs à un ID d'opération spécifique :

```
aws cloudformation describe-events \
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57
```

Pour afficher uniquement les événements ayant échoué à des fins de débogage, utilisez le `--filter FailedEvents=true` paramètre :

```
aws cloudformation describe-events \
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57 \
  --filter FailedEvents=true
```

La nouvelle fonctionnalité de filtrage des identifiants d'opération vous permet de vous concentrer sur des opérations spécifiques et les événements connexes. Cela est particulièrement utile pour :

- Résolution de défaillances spécifiques : isolez les événements liés à un échec d'une opération pour comprendre ce qui s'est mal passé.
- Surveillance des opérations de longue durée : suivez la progression des mises à jour complexes ou des déploiements à grande échelle.
- Audit des modifications : passez en revue tous les événements associés à une opération de mise à jour particulière.
- Analyse des causes premières : utilisez le filtre d'échec pour identifier rapidement la source des problèmes de déploiement.

Codes d'état de la pile

Le tableau suivant décrit les champs renvoyés par la `describe-events` commande lors de l'utilisation du filtrage par ID d'opération :

Champ	Description
EventId	Identifiant de l'événement.
OperationId	Identifiant unique pour l'opération qui a généré cet événement.
StackName	Nom de la pile à laquelle l'événement correspond.
StackId	Identifiant de la pile à laquelle l'événement correspond.
LogicalResourceId	Identifiant logique de la ressource.
PhysicalResourceId	Identifiant physique de la ressource.
ResourceProperties	Propriétés de la ressource.
ResourceType	Type de ressource.
Timestamp	Moment où l'événement est survenu.
ResourceStatus	État de la ressource (CREATE_COMPLETE ,UPDATE_FAILED , etc.).
DetailedStatus	État détaillé de la pile. Si CONFIGURATION_COMPLETE est présent, la phase de configuration des ressources de la pile est terminée et la stabilisation des ressources est en cours.
ResourceStatusReason	Plus d'informations sur cet état.

Afficher la chronologie d'un déploiement de CloudFormation stack

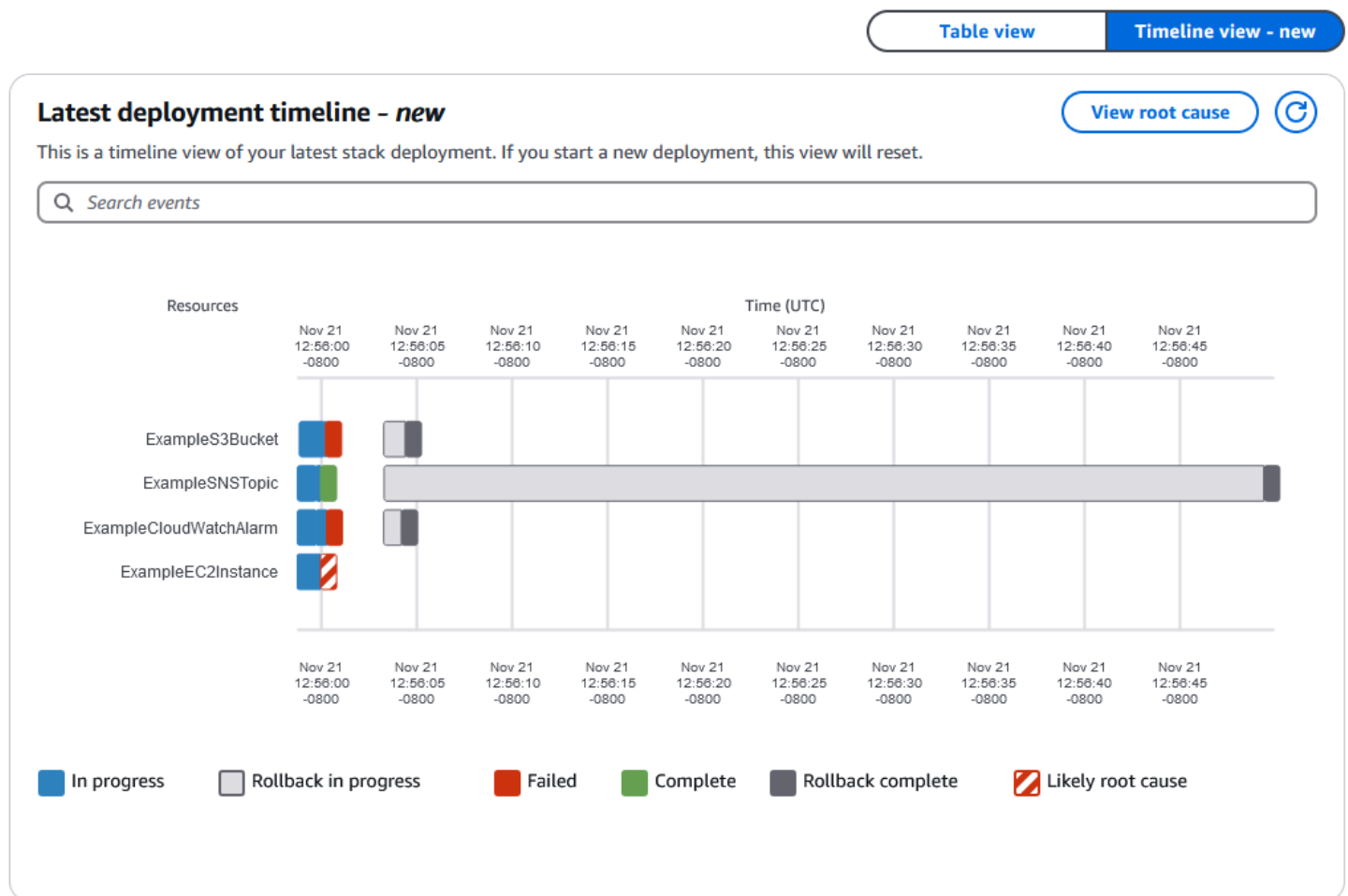
Le graphique de la chronologie du déploiement de la pile fournit une représentation visuelle de la chronologie du déploiement de la pile. Cette vue affiche les statuts de déploiement de la pile et de chacune de ses ressources, ainsi que les heures auxquelles chaque statut a changé. Les statuts de déploiement de la pile sont représentés par une couleur correspondante.

Rubriques

- [Comprendre le graphique de la chronologie du déploiement de la pile](#)
- [Affichage du graphique chronologique du déploiement de la pile \(console\)](#)

Comprendre le graphique de la chronologie du déploiement de la pile

L'image suivante montre le graphique chronologique d'un déploiement de stack qui a échoué en raison de l'échec du lancement d'une ressource d' EC2 instance Amazon.



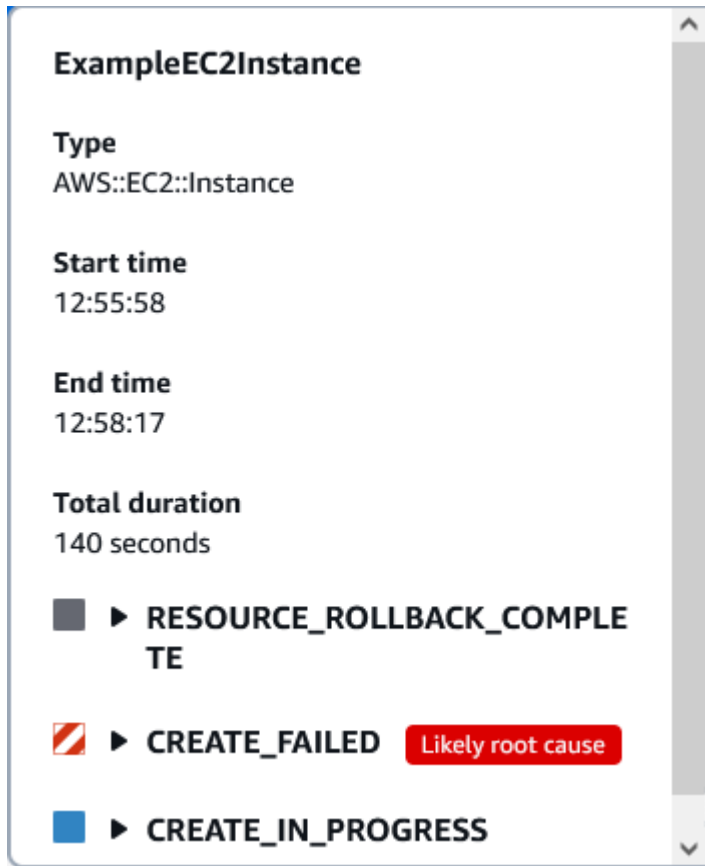
Les noms des ressources de la pile se trouvent sur le côté gauche du graphique, et la date et l'heure relatives aux temps de déploiement se trouvent en haut du graphique.

Chaque ressource commence avec le statut En cours. La barre d'état passe à Terminé pour chaque déploiement réussi. La barre d'état passe à Échec lorsqu'une ressource ne parvient pas à se déployer. Lorsqu'une ressource ne parvient pas à se déployer et que le déploiement de la pile

échoue également, la ressource responsable de l'échec du déploiement de la pile reçoit le statut Échec racine probable.

Après l'échec de l'opération de déploiement de la pile, la ressource déployée avec succès commence à être restaurée et passe au statut Restauration en cours. Les statuts passent à Restauration terminée une fois que la ressource a fini d'être restaurée.

En sélectionnant chaque ressource, vous obtenez des informations plus détaillées sur le calendrier de déploiement :



En sélectionnant une ressource, vous pouvez voir le Type, l'Heure de début du déploiement, l'Heure de fin du déploiement et la Durée totale du déploiement. Vous trouverez également l'Heure de début, l'Heure de fin et la Durée de chaque statut de déploiement dans les menus déroulants ci-dessous. Si la ressource n'a pas pu être déployée, une Raison de l'échec sera fournie.

Pour plus d'informations sur les statuts des piles, consultez [Codes d'état de la pile](#).

Affichage du graphique chronologique du déploiement de la pile (console)

Pour afficher le graphique chronologique du déploiement de la pile :

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le nom dans Région AWS lequel vous avez créé la pile.
3. Sur la page Stacks de la CloudFormation console, choisissez le nom de la pile. CloudFormation affiche les détails de la pile sélectionnée.
4. Choisissez l'onglet Événements pour afficher les événements CloudFormation de pile générés pour votre pile.
5. Sélectionnez le bouton Graphique chronologique pour afficher le graphique chronologique de votre pile.

Comprendre les événements de création de CloudFormation piles

Pendant le déploiement de la pile, plusieurs événements se produisent pour créer, configurer et valider les ressources définies dans le modèle de pile. La compréhension de ces événements peut vous aider à optimiser votre processus de création de pile et à rationaliser les déploiements.

- Événements de création de ressources : lorsque chaque ressource démarre le processus de création, un événement Statut de type `CREATE_IN_PROGRESS` est défini. Cet événement indique que la ressource est en cours de provisionnement.
- Vérification de la cohérence finale : une partie importante du temps de création de la pile est consacrée à la vérification de la cohérence finale des ressources créées par la pile. Au cours de cette phase, le service effectue des contrôles de cohérence internes, s'assurant que la ressource est pleinement opérationnelle et répond aux critères de stabilisation du service définis par chacun Service AWS.
- Événement de configuration terminée : lorsque chaque ressource a terminé la phase de vérification de la cohérence finale du provisionnement, un Statut détaillé de l'événement `CONFIGURATION_COMPLETE` est défini.
- Événement de création de ressource terminée : une fois que la ressource a été créée et configurée comme spécifié, et que la configuration correspond à celle spécifiée dans le modèle, l'événement Statut de `CREATE_COMPLETE` est défini.

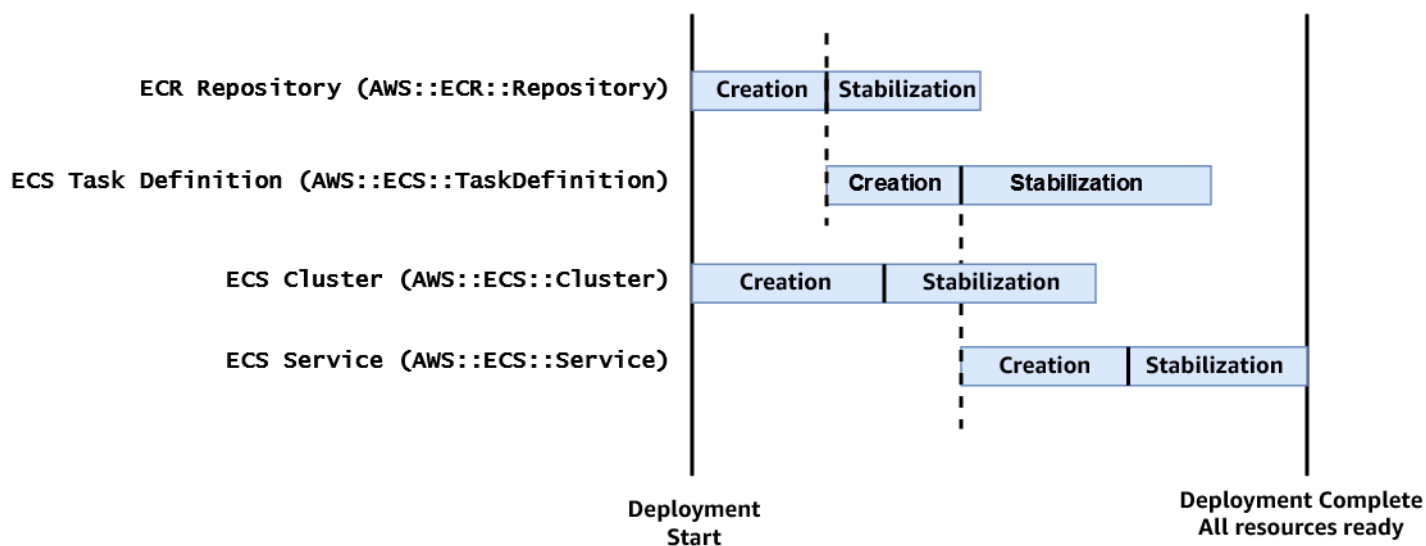
Vous pouvez utiliser l'événement `CONFIGURATION_COMPLETE` pour rationaliser votre processus de création de pile dans les scénarios où la vérification de la cohérence finale des ressources n'est pas nécessaire, comme la validation d'une configuration de pile de préproduction ou le provisionnement inter-piles. Cet événement peut être utilisé de différentes manières. Par exemple, vous pouvez

l'utiliser comme signal visuel pour éviter d'attendre la fin de la vérification de la cohérence des ressources ou de la pile. Vous pouvez également l'utiliser pour créer un mécanisme automatisé à l'aide de l'intégration continue et de la livraison continue (CI/CD) afin de déclencher des actions supplémentaires.

⚠ Important

Bien que l'utilisation de l'événement `CONFIGURATION_COMPLETE` accélère la création des piles, il est important de connaître ses inconvénients. Tout d'abord, il n'est pris en charge que pour un sous-ensemble de types de ressources qui prennent en charge la détection des écarts. Pour obtenir la liste des types de ressources qui prennent en charge la détection des écarts, consultez [Prise en charge des types de ressources](#). Cette approche peut ne pas convenir à tous les scénarios, en particulier lorsque les ressources nécessitent des vérifications approfondies de la cohérence finale afin de garantir une disponibilité opérationnelle totale dans l'environnement cloud (par exemple, dans les environnements de production). Nous vous recommandons d'évaluer soigneusement vos exigences de déploiement et l'importance des contrôles de cohérence pour chaque ressource. Utilisez l'événement `CONFIGURATION_COMPLETE` pour optimiser les vitesses de déploiement sans compromettre l'intégrité et la fiabilité de votre infrastructure.

Étant donné que la définition de l'événement `CONFIGURATION_COMPLETE` n'est pas garantie, tous les scénarios qui l'utilisent doivent être prêts à gérer un événement `CREATE_COMPLETE` lorsqu'aucun événement `CONFIGURATION_COMPLETE` n'a été défini.



Lorsque le déploiement de la pile commence, les ressources `AWS::ECR::Repository` et `AWS::ECS::Cluster` lancent toutes deux le processus de création (`ResourceStatus = CREATE_IN_PROGRESS`). Lorsque le type de ressource `AWS::ECR::Repository` a lancé la vérification de cohérence finale (`DetailedStatus = CONFIGURATION_COMPLETE`), la ressource `AWS::ECS::TaskDefinition` peut alors lancer le processus de création. De même, une fois que la ressource `AWS::ECS::TaskDefinition` a lancé la vérification de cohérence finale, la ressource `AWS::ECS::Service` lance le processus de création.

Événements **CREATE_IN_PROGRESS** et **CREATE_COMPLETE**

- [Stack] : **CREATE_IN_PROGRESS**
- [Resource] : référentiel ECR **CREATE_IN_PROGRESS**
- [Resource] : cluster ECS **CREATE_IN_PROGRESS**
- [Resource] : référentiel ECR **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource] : définition de tâche ECS **CREATE_IN_PROGRESS**
- [Resource] : cluster ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource] : définition de tâche ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource] : service ECS **CREATE_IN_PROGRESS**
- [Resource] : référentiel ECR **CREATE_COMPLETE**
- [Resource] : cluster ECS **CREATE_COMPLETE**
- [Resource] : service ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Stack] : **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Resource] : définition de tâche ECS **CREATE_COMPLETE**
- [Resource] : service ECS **CREATE_COMPLETE**
- [Stack] : **CREATE_COMPLETE**

Surveiller la progression d'une mise à jour de pile

Pour suivre la progression d'une mise à jour de la pile, vous pouvez consulter ses événements. L'onglet Événements de la pile affiche chaque étape importante de la création et de la mise à jour de la pile, triées par heure de chaque événement, les événements les plus récents apparaissant en haut. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Rubriques

- [Événements générés lors d'une mise à jour de pile réussie](#)
- [Événements générés en cas d'échec d'une mise à jour de ressource](#)

Événements générés lors d'une mise à jour de pile réussie

Le début du processus de mise à jour de la pile est marqué d'un événement `UPDATE_IN_PROGRESS` pour la pile :

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

Viennent ensuite les événements qui marquent le début et la fin de la mise à jour de chaque ressource qui a été modifiée dans le modèle de mise à jour. Par exemple, la mise à jour d'une ressource [AWS::RDS::DBInstance](#) nommée MyDB entraînerait les entrées suivantes :

```
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

L'`UPDATE_IN_PROGRESS` événement est enregistré lorsqu'il est CloudFormation signalé que la ressource a commencé à être mise à jour. L'événement `UPDATE_COMPLETE` est enregistré lorsque la ressource a été créée avec succès.

Une CloudFormation fois la pile mise à jour réussie, vous verrez l'événement suivant :

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

Important

Lors des opérations de mise à jour de la pile, s'il est CloudFormation nécessaire de remplacer une ressource existante, il crée d'abord une nouvelle ressource, puis supprime l'ancienne ressource. Cependant, dans certains cas, il CloudFormation peut être impossible de supprimer l'ancienne ressource (par exemple, si l'utilisateur n'est pas autorisé à supprimer une ressource d'un type donné).

CloudFormation fait trois tentatives pour supprimer l'ancienne ressource. S'il n'est pas CloudFormation possible de supprimer l'ancienne ressource, elle est supprimée de la

pile et continue de mettre à jour la pile. Lorsque la mise à jour de la pile est terminée, CloudFormation un événement de `UPDATE_COMPLETE` pile est émis, mais inclut un événement indiquant `StatusReason` qu'une ou plusieurs ressources n'ont pas pu être supprimées. CloudFormation émet également un `DELETE_FAILED` événement pour la ressource spécifique, avec un correspondant `StatusReason` fournissant plus de détails sur les raisons de l' CloudFormation échec de la suppression de la ressource.

L'ancienne ressource existe toujours et continuera d'être facturée, mais elle n'est plus accessible via CloudFormation. Pour supprimer l'ancienne ressource, accédez à l'ancienne ressource directement à l'aide de la console ou de l'API du service sous-jacent.

Ceci est également vrai pour les ressources que vous avez retirées du modèle de pile et qui seront donc supprimées de la pile lors de la mise à jour de la pile.

Événements générés en cas d'échec d'une mise à jour de ressource

Si la mise à jour d'une ressource échoue, CloudFormation signale un `UPDATE_FAILED` événement indiquant la raison de l'échec. Par exemple, si votre modèle de mise à jour spécifie une modification de propriété qui n'est pas prise en charge par la ressource, telle que la réduction de la taille de `AllocatedStorage` pour une ressource [AWS::RDS::DBInstance](#), vous verrez des événements tels que ceux-ci :

```
2011-09-30 09:36 PDT AWS::RDS::DBInstance MyDB UPDATE_FAILED Size cannot be less than
current size; requested: 5; current: 10
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

En cas d'échec d'une mise à jour CloudFormation des ressources, restaure toutes les ressources mises à jour lors de la mise à niveau vers leurs configurations antérieures à la mise à jour. Voici un exemple d'événements que vous pouvez voir en cas d'annulation d'une mise à jour :

```
2011-09-30 09:38 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_COMPLETE
2011-09-30 09:38 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:37 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
2011-09-30 09:37 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_IN_PROGRESS The
following resource(s) failed to update: [MyDB]
```


Poursuite de la restauration d'une mise à jour

Parfois, lorsque vous CloudFormation essayez d'annuler une mise à jour de la pile, il ne peut pas annuler toutes les modifications apportées au cours du processus de mise à jour. C'est ce qu'on appelle l'état `UPDATE_ROLLBACK_FAILED`. Par exemple, vous pouvez avoir une pile qui commence à revenir à une ancienne instance de base de données supprimée en dehors de CloudFormation. Comme CloudFormation ne sait pas que la base de données a été supprimée, il suppose qu'elle existe toujours et tente de l'utiliser pour effectuer la restauration, ce qui entraîne l'échec de l'opération.

Une pile dans l'état `UPDATE_ROLLBACK_FAILED` ne peut pas être mise à jour, mais elle peut être restaurée à un état fonctionnel (`UPDATE_ROLLBACK_COMPLETE`). Après avoir rétabli les paramètres d'origine de la pile, vous pouvez essayer de la mettre à jour à nouveau.

Dans la plupart des cas, vous devez corriger l'erreur qui entraîne l'échec de la restauration de la mise à jour avant de pouvoir continuer à restaurer la pile. Dans d'autres cas, vous pouvez poursuivre la restauration de la mise à jour sans aucune modification (par exemple, lorsqu'une opération de la pile arrive à expiration).

Note

Si vous utilisez des piles imbriquées, la restauration de la pile parent tente de restaurer également toutes les piles enfants.

Pour poursuivre la restauration d'une mise à jour (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de la pile.
3. Sur la page Piles, sélectionnez la pile que vous voulez mettre à jour, sélectionnez Actions de pile, puis sélectionnez Continuer la restauration de la mise à jour.

Si aucune des solutions ne [Résolution des erreurs](#) fonctionne, vous pouvez utiliser l'option avancée pour ignorer les ressources qui ne CloudFormation peuvent pas être restaurées avec succès. Vous devez [rechercher](#) et saisir la logique IDs des ressources que vous souhaitez

ignorer. Spécifiez uniquement les ressources qui sont passées à l'état `UPDATE_FAILED` pendant l'état `UpdateRollback` et non pendant la mise à jour directe.

 Warning

CloudFormation définit le statut des ressources spécifiées sur `UPDATE_COMPLETE` et continue de faire reculer la pile. Une fois la restauration terminée, l'état des ressources ignorées ne correspond pas à celui des ressources contenues dans le modèle de pile. Avant de procéder à une autre mise à jour de pile, vous devez mettre à jour la pile ou les ressources de façon à les rendre cohérentes les unes par rapport aux autres. A défaut, les futures mises à jour de la pile risquent d'échouer et la pile deviendra irrécupérable.

Spécifiez le nombre minimum de ressources requises pour restaurer votre pile avec succès. Par exemple, une mise à jour de ressource ayant échoué peut entraîner l'échec de ressources dépendantes. Dans ce cas, il n'est peut-être pas nécessaire d'ignorer les ressources dépendantes.

Pour ignorer des ressources faisant partie de piles imbriquées, utilisez le format suivant : ***NestedStackName.ResourceLogicalID***. Si vous souhaitez spécifier l'ID logique d'une ressource de pile (Type: `AWS::CloudFormation::Stack`) dans la liste `ResourcesToSkip`, sa ressource de pile imbriquée correspondante doit être à l'un des états suivants : `DELETE_IN_PROGRESS`, `DELETE_COMPLETE` ou `DELETE_FAILED`.

Pour poursuivre la restauration d'une mise à jour (AWS CLI)

- Utilisez la commande [continue-update-rollback](#) avec l'option `--stack-name` pour spécifier l'ID de la pile que vous voulez continuer à restaurer.

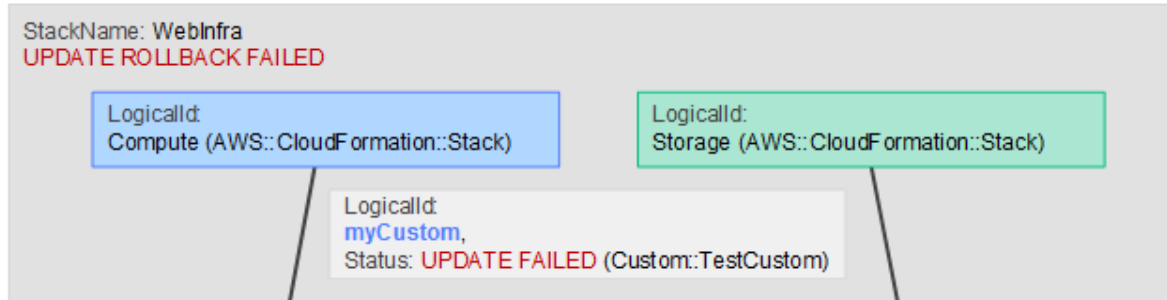
Continuer la restauration à partir des mises à jour de piles imbriquées ayant échoué

Lorsque vous avez plusieurs piles imbriquées les unes dans les autres, vous devrez peut-être ignorer les ressources sur plusieurs niveaux imbriqués pour que la pile complète revienne à un état fonctionnel.

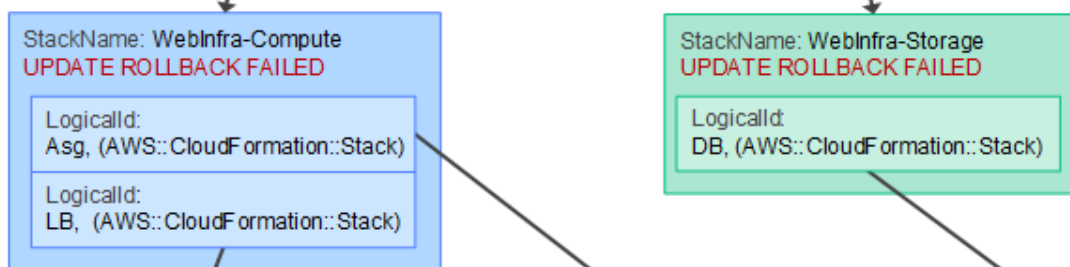
Par exemple, vous avez une pile racine appelée WebInfra qui contient deux piles plus petites à l'intérieur : WebInfra-Compute et WebInfra-Storage. Ces deux piles ont également leurs propres piles imbriquées à l'intérieur.

Si un problème survient pendant une mise à jour et que le processus de mise à jour échoue, toute la hiérarchie de la pile peut se retrouver dans l'état UPDATE_ROLLBACK_FAILED, comme illustré dans le diagramme suivant.

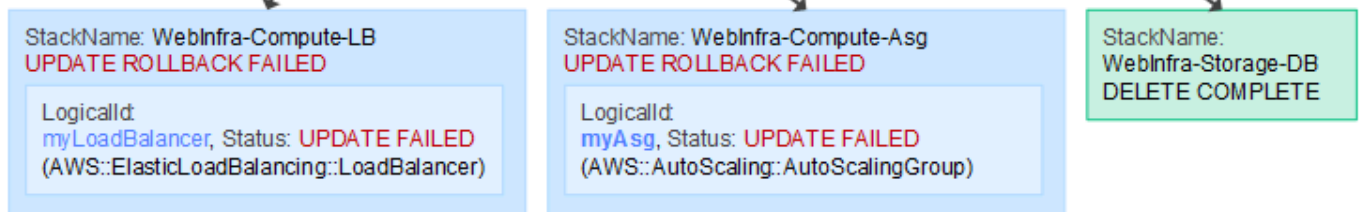
Root



Level 1



Level 2



Note

Les noms de pile de cet exemple sont tronqués par souci de simplification. Les noms des piles d'enfants sont généralement générés par CloudFormation et contiennent des chaînes aléatoires uniques, de sorte que les noms réels peuvent ne pas être faciles à utiliser.

Pour que la pile racine redevienne opérationnelle à l'aide de la commande `continue-update-rollback`, vous devez utiliser l'option `--resources-to-skip` pour ignorer les ressources dont la restauration a échoué.

L'exemple `continue-update-rollback` suivant reprend une opération de restauration suite à un échec antérieur de la mise à jour de la pile. Dans cet exemple, l'option `--resources-to-skip` inclut les éléments suivants :

- *myCustom*
- *WebInfra-Compute-Asg.myAsg*
- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

Pour les ressources de la pile racine, il suffit de fournir l'ID logique, par exemple *myCustom*. Cependant, pour les ressources contenues dans des piles imbriquées, vous devez fournir à la fois le nom de la pile imbriquée et son ID logique, séparés par un point. Par exemple, *WebInfra-Compute-Asg.myAsg*.

```
aws cloudformation continue-update-rollback --stack-name WebInfra \
  --resources-to-skip myCustom WebInfra-Compute-Asg.myAsg WebInfra-Compute-
  LB.myLoadBalancer WebInfra-Storage.DB
```

Pour trouver le nom d'une pile imbriquée

Vous pouvez le localiser dans l'ID de pile ou le nom de ressource Amazon Resource Name (ARN) de la pile enfant.

L'exemple d'ARN suivant fait référence à une pile nommée `WebInfra-Storage-Z2VKC706XKXT`.

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

Pour trouver l'ID logique d'une pile imbriquée

Vous pouvez trouver l'ID logique d'une pile enfant dans la définition de modèle de son parent. Dans le diagramme, le `LogicalId` de la pile enfant `WebInfra-Storage-DB` est `DB` dans sa pile parent `WebInfra-Storage`.

Dans la CloudFormation console, vous pouvez également trouver l'identifiant logique dans la colonne ID logique de la ressource de pile sous l'onglet Ressources ou dans l'onglet Événements. Pour de plus amples informations, veuillez consulter [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

Détermination de la cause d'un échec de pile

Si la création de votre pile échoue, cela CloudFormation peut vous aider à déterminer l'événement susceptible d'être à l'origine de la défaillance de la pile. Selon le scénario et vos autorisations, les événements AWS CloudTrail peuvent fournir des détails supplémentaires sur la cause profonde si la Raison de l'état indiquée dans Événements n'est pas claire.

Détermination de la cause profonde d'un échec de pile

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Piles, sélectionnez la pile échouée.
3. Sélectionnez l'onglet Événements.
4. Choisissez Détecter la cause première. CloudFormation analysera la panne et indiquera l'événement qui en est probablement la cause en ajoutant une étiquette « cause première probable » à l'état de l'événement spécifique. Reportez-vous à la section Motif du statut pour plus d'explications sur l'état de la CloudFormation console.
5. Sélectionner l'État d'échec portant l'étiquette Cause profonde probable afin d'obtenir plus d'informations sur la cause de l'échec. En fonction du scénario et de vos autorisations, vous pourrez peut-être consulter un CloudTrail événement détaillé. Résultats possibles après sélection de l'État
 - CloudTrail des événements liés à ce problème sont disponibles et peuvent aider à le résoudre. Afficher CloudTrail les événements.
 - Nous n'avons trouvé aucun CloudTrail événement lié à ce problème susceptible d'aider à le résoudre.
 - Vos autorisations actuelles ne vous permettent pas d'accéder à l'affichage CloudTrail des événements. [En savoir plus](#).
 - Au cours du processus de vérification des CloudTrail événements disponibles, revenez dans quelques minutes.
 - Une erreur s'est produite lors de la récupération des CloudTrail événements. Pour une inspection manuelle, rendez-vous sur la CloudTrail console.

6. Si le motif indiqué dans le champ Motif du statut n'est pas clair et que la cause première affiche un lien vers la CloudTrail console, ouvrez le lien pour afficher l'événement afin de trouver une cause première détaillée.

Pour plus d'informations sur les CloudTrail événements, voir [Comprendre les CloudTrail événements](#) et [le contenu des CloudTrail enregistrements](#).

Pour plus d'informations sur l'historique des CloudTrail événements, voir [Utilisation de l'historique des CloudTrail événements](#).

 Note

Les piles imbriquées ne prennent pas en charge la fonction Détecter la cause profonde.

Choisissez comment gérer les défaillances lors de l'approvisionnement des ressources

Si votre opération de stack échoue, vous n'avez pas à annuler les ressources déjà correctement provisionnées et à recommencer depuis le début à chaque fois. Au lieu de cela, vous pouvez dépanner les ressources dans un état `CREATE_FAILED` ou `UPDATE_FAILED`, puis reprendre le provisionnement à partir du point où le problème s'est produit.

Pour ce faire, vous devez activer l'option Conserver les ressources correctement provisionnées. Cette option est disponible pour tous les déploiements de pile et les opérations d'ensembles de modifications.

- Pour la création de piles, si vous choisissez l'option Conserver les ressources correctement provisionnées, CloudFormation préserve l'état des ressources créées avec succès et laisse les ressources défaillantes dans un état d'échec jusqu'à ce que la prochaine opération de mise à jour soit effectuée.
- Lors des opérations de mise à jour et de modification, choisir Préserver les ressources provisionnées avec succès permet de conserver l'état des ressources qui ont été provisionnées avec succès tout en rétablissant les ressources qui ont échoué à leur dernier état stable connu. Les ressources en échec se trouveront à l'état `UPDATE_FAILED`. Les ressources sans dernier état stable connu seront supprimées lors de la prochaine opération de pile.

Rubriques

- [Présentation des options d'échec de pile](#)
- [Conditions requises pour suspendre la restauration de la pile](#)
- [Préserver les ressources correctement allouées \(console\)](#)
- [Préserver les ressources correctement allouées \(AWS CLI\)](#)

Présentation des options d'échec de pile

Avant d'exécuter une opération depuis la CloudFormation console, l'API ou AWS CLI, spécifiez le comportement en cas de défaillance des ressources provisionnées. Passez ensuite au processus de déploiement de vos ressources sans aucune autre modification. En cas de défaillance opérationnelle, CloudFormation s'arrête à la première défaillance de chaque chemin de provisionnement indépendant. CloudFormation identifie les dépendances entre les ressources afin de paralléliser les actions de provisionnement indépendantes. Il procède ensuite à l'allocation des ressources sur chaque chemin d'allocation indépendant jusqu'à ce qu'il rencontre un échec. La défaillance d'un chemin n'affecte pas les autres chemins de provisionnement. CloudFormation continuera à fournir les ressources jusqu'à leur achèvement ou s'arrêtera en cas de panne différente.

Corrigez les éventuels problèmes pour poursuivre le processus de déploiement. CloudFormation effectue les mises à jour nécessaires avant de réessayer des actions de provisionnement sur des ressources qui n'ont pas pu être correctement provisionnées plus tôt. Vous résolvez les problèmes en soumettant les opérations Retry (Réessayer), Update (Mettre à jour) ou Roll back (Restaurer). Par exemple, si vous allouez une instance Amazon EC2 et que l'instance EC2 échoue lors d'une opération de création, vous pouvez rechercher l'erreur, plutôt que de restaurer immédiatement la ressource ayant échoué. Vous pouvez passer en revue les contrôles d'état du système et les contrôles d'état des instances, puis sélectionner l'opération Retry (Réessayer) une fois que les problèmes sont résolus.

Lorsqu'une opération de pile échoue et que vous avez spécifié *Preserve successfully provisioned resources* (Préserver les ressources correctement allouées) à partir du menu *Stack failure options* (Options d'échec de pile), vous pouvez sélectionner les options suivantes.

- **Retry (Réessayer)** : relance l'opération d'allocation sur les ressources en échec et continue d'allouer le modèle jusqu'à la fin de l'opération de pile ou jusqu'à l'échec suivant. Sélectionnez cette option si la ressource n'a pas pu être provisionnée en raison d'un problème ne nécessitant pas de modification du modèle, tel qu'une autorisation Gestion des identités et des accès AWS (IAM).

- **Update (Mettre à jour)** : les ressources qui ont été allouées sont mises à jour lors des mises à jour des modèles. Les ressources qui n'ont pas pu être créées ou mises à jour seront relancées. Sélectionnez cette option si la ressource n'a pas pu être allouée en raison d'erreurs de modèle et si vous avez modifié le modèle. Lorsque vous mettez à jour une pile qui se trouve à l'état **FAILED**, vous devez sélectionner **Preserve successfully provisioned resources** (Préserver les ressources correctement allouées) pour **Stack failure options** (Options d'échec de pile) afin de poursuivre la mise à jour de votre pile.
- **Annulation** : CloudFormation ramène la pile au dernier état stable connu.

Conditions requises pour suspendre la restauration de la pile

Pour éviter CloudFormation de revenir en arrière et de supprimer automatiquement les ressources créées avec succès, les conditions suivantes doivent être remplies.

1. Lorsque vous créez ou mettez à jour la pile, vous devez choisir l'option **Préserver les ressources correctement provisionnées**. Cela indique à CloudFormation de ne pas supprimer les ressources créées avec succès, même si l'opération globale de pile échoue.
2. L'opération de pile doit avoir échoué, ce qui signifie que l'état de la pile est soit **CREATE_FAILED** soit **UPDATE_FAILED**.

Note

Les types de mise à jour immuables ne sont pas pris en charge.

Préserver les ressources correctement allouées (console)

Create stack

Pour préserver les ressources correctement allouées lors d'une opération de création de pile

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Depuis la page CloudFormation, choisissez **Piles**, puis **Créer une pile avec de nouvelles ressources** (standard).
3. Pour la Condition préalable – Préparer le modèle, sélectionnez **Choisir un modèle existant**.

4. Sous Spécifier le modèle, choisissez soit de spécifier l'URL du compartiment S3 qui contient votre modèle de pile, soit de charger un fichier de modèle de pile. Ensuite, choisissez Suivant.
5. Sur la page Spécifier les détails de la pile, saisissez un nom pour la pile dans la zone Stack name (Nom de la pile).
6. Dans la section Paramètres, spécifiez les paramètres définis dans votre modèle de pile.

Vous pouvez utiliser ou modifier n'importe quel paramètre contenant des valeurs par défaut.

7. Lorsque vous êtes satisfait de vos paramètres, choisissez Next (Suivant).
8. Sur la page Configurer les options de la pile, vous pouvez définir des options supplémentaires pour votre pile.
9. Pour Stack failure options (Options d'échec de pile), sélectionnez Preserve successfully provisioned resources (Préserver les ressources correctement allouées).
10. Lorsque vous êtes satisfait de vos options de pile, choisissez Next (Suivant).
11. Passez en revue votre pile sur la page Révision et sélectionnez Create stack (Créer une pile).

Résultats : les ressources qui n'ont pas pu être créées font passer l'état de la pile à `CREATE_FAILED` pour empêcher la restauration de la pile lorsque l'opération de pile échoue. Les ressources qui sont correctement allouées se trouvent à l'état `CREATE_COMPLETE`. Vous pouvez surveiller la pile dans l'onglet Stack events (Événements de la pile).

Update stack

Pour préserver les ressources correctement allouées lors d'une opération de mise à jour de pile

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sélectionnez la pile que vous souhaitez mettre à jour, puis cliquez sur Update (Mettre à jour).
3. Sur la page Mettre à jour une pile, choisissez un modèle de pile en utilisant l'une des options suivantes :
 - Utiliser le modèle existant
 - Replace current template (Remplacer le modèle actuel)
 - Modifier le modèle dans Infrastructure Composer

Acceptez vos paramètres et sélectionnez Next (Suivant).

4. Sur la page Définir les détails de la pile, spécifiez les paramètres définis dans votre modèle de pile.

Vous pouvez utiliser ou modifier n'importe quel paramètre contenant des valeurs par défaut.

5. Lorsque vous êtes satisfait de vos paramètres, choisissez Next (Suivant).
6. Sur la page Configurer les options de la pile, vous pouvez définir des options supplémentaires pour votre pile.
7. Pour Comportement en cas d'échec de l'allocation, sélectionnez Preserve successfully provisioned resources (Préserver les ressources correctement allouées).
8. Lorsque vous êtes satisfait de vos options de pile, choisissez Next (Suivant).
9. Passez en revue votre pile sur la page Révision et sélectionnez Update stack (Mettre à jour une pile).

Résultats : les ressources qui n'ont pas pu être mises à jour passent l'état de la pile à UPDATE_FAILED et reviennent au dernier état stable connu. Les ressources dont le dernier état stable est connu seront supprimées CloudFormation lors de la prochaine opération de pile. Les ressources qui sont correctement allouées se trouvent à l'état CREATE_COMPLETE ou UPDATE_COMPLETE. Vous pouvez surveiller la pile dans l'onglet Stack events (Événements de la pile).

Change set

Note

Vous pouvez initier un jeu de modifications pour une pile dont l'état est CREATE_FAILED ou UPDATE_FAILED, mais pas pour un état UPDATE_ROLLBACK_FAILED.

Pour préserver les ressources correctement allouées lors d'une opération de modification de pile

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sélectionnez la pile qui contient le jeu de modifications que vous souhaitez initier, puis choisissez l'onglet Change sets (Jeux de modifications).
3. Sélectionnez le jeu de modifications, puis cliquez sur Execute (Exécuter).
4. Pour Execute change set (Exécuter le jeu de modifications), sélectionnez l'option Preserve successfully provisioned resources (Préserver les ressources correctement allouées).

5. Sélectionnez Execute change set (Exécuter le jeu de modifications).

Résultats : les ressources qui n'ont pas pu être mises à jour passent l'état de la pile à `UPDATE_FAILED` et reviennent au dernier état stable connu. Les ressources dont le dernier état stable est connu seront supprimées CloudFormation lors de la prochaine opération de pile. Les ressources qui sont correctement allouées se trouvent à l'état `CREATE_COMPLETE` ou `UPDATE_COMPLETE`. Vous pouvez surveiller la pile dans l'onglet Stack events (Événements de la pile).

Préserver les ressources correctement allouées (AWS CLI)

Create stack

Pour préserver les ressources correctement allouées lors d'une opération de création de pile

Spécifiez l'option `--disable-rollback` ou l'énumération on-failure `DO_NOTHING` au cours d'une opération [create-stack](#).

1. Fournissez un nom de pile et un modèle à la commande `create-stack` avec l'option `--disable-rollback`.

```
aws cloudformation create-stack --stack-name myteststack \  
  --template-body file://template.yaml \  
  --disable-rollback
```

La commande renvoie le résultat suivant.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. Décrivez l'état de la pile à l'aide de la commande `describe-stacks`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

La commande renvoie le résultat suivant.

```
{
```

```

    "Stacks": [
      {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "Description": "AWS CloudFormation Sample Template",
        "Tags": [],
        "Outputs": [],
        "StackStatusReason": "The following resource(s) failed to create:
[MyBucket]",
        "CreationTime": "2013-08-23T01:02:15.422Z",
        "Capabilities": [],
        "StackName": "myteststack",
        "StackStatus": "CREATE_FAILED",
        "DisableRollback": true
      }
    ]
  }

```

Update stack

Pour préserver les ressources provisionnées avec succès pendant une opération de mise à jour de la pile

1. Fournissez un nom de pile et un modèle existants à la commande `update-stack` avec l'option `--disable-rollback`.

```

aws cloudformation update-stack --stack-name myteststack \
  --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template
  --disable-rollback

```

La commande renvoie le résultat suivant.

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}

```

2. Décrivez l'état de la pile à l'aide de la commande `describe-stacks` ou `describe-stack-events`.

```

aws cloudformation describe-stacks --stack-name myteststack

```

La commande renvoie le résultat suivant.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "UPDATE_COMPLETE",
      "DisableRollback": true
    }
  ]
}
```

Change set

Note

Vous pouvez initier un jeu de modifications pour une pile dont l'état est `CREATE_FAILED` ou `UPDATE_FAILED`, mais pas pour un état `UPDATE_ROLLBACK_FAILED`.

Pour préserver les ressources correctement allouées lors d'une opération de modification de pile

Spécifiez l'option `--disable-rollback` lors d'une [execute-change-set](#) opération.

1. Fournissez un nom de pile et un modèle à la commande `execute-change-set` avec l'option `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --template-body file://template.yaml
```

La commande renvoie le résultat suivant.

```
{
  "Id": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. Lancez le jeu de modifications à l'aide de l'option `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --disable-rollback
```

3. Déterminez l'état de la pile à l'aide de la commande `describe-stacks` ou `describe-stack-events`.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

La commande renvoie le résultat suivant.

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "EventId": "49c966a0-7b74-11ea-8071-024244bb0672",
      "StackName": "myteststack",
      "LogicalResourceId": " MyBucket",
      "PhysicalResourceId": "myteststack-MyBucket-abcdefghijkl1",
      "ResourceType": "AWS::S3::Bucket",
      "Timestamp": "2020-04-10T21:43:17.015Z",
      "ResourceStatus": "UPDATE_FAILED"
      "ResourceStatusReason": "User XYZ is not allowed to perform
S3::UpdateBucket on MyBucket"
    }
  ]
}
```

4. Corrigez les erreurs d'autorisation et réessayez l'opération.

```
aws cloudformation update-stack --stack-name myteststack \
  --use-previous-template --disable-rollback
```

La commande renvoie le résultat suivant.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

5. Décrivez l'état de la pile à l'aide de la commande `describe-stacks` ou `describe-stack-events`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

La commande renvoie le résultat suivant.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "UPDATE_COMPLETE",
      "DisableRollback": true
    }
  ]
}
```

Rétablissement d'une pile

Vous pouvez utiliser la commande [rollback-stack](#) pour restaurer une pile avec un état `CREATE_FAILED` ou `UPDATE_FAILED` pile à son dernier état stable.

La commande `rollback-stack` suivante annule la pile spécifiée.

```
aws cloudformation rollback-stack --stack-name myteststack
```

La commande renvoie le résultat suivant.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Note

L'opération `rollback-stack` supprimera une pile si elle ne contient pas un dernier état stable connu.

Réduisez votre batterie en CloudFormation cas de violation d'alarme grâce à des déclencheurs

Grâce aux déclencheurs d'annulation, vous pouvez CloudFormation surveiller l'état de votre application lors de la création et de la mise à jour de la pile, puis annuler cette opération si l'application dépasse le seuil d'alarmes que vous avez spécifié. Pour chaque déclencheur d'annulation que vous créez, vous spécifiez l' CloudWatch alarme qui CloudFormation doit être surveillée. CloudFormation surveille les alarmes spécifiées pendant l'opération de création ou de mise à jour de la pile, et pendant la durée spécifiée après le déploiement de toutes les ressources. Si l'une des alarmes se déclenche pendant ALARM le fonctionnement de la pile ou pendant la période de surveillance, CloudFormation annule l'ensemble du fonctionnement de la pile.

Vous pouvez définir une durée de surveillance comprise entre la valeur par défaut 0 et 180 minutes. Pendant cette période, CloudFormation surveille tous les déclencheurs de restauration après que l'opération de création ou de mise à jour de la pile a déployé toutes les ressources nécessaires. Si l'une des alarmes se déclenche pendant ALARM le fonctionnement de la pile ou pendant cette période de surveillance, CloudFormation annule l'ensemble du fonctionnement de la pile. Ensuite, pour les opérations de mise à jour, si la période de surveillance expire sans qu'aucune alarme ne ALARM se déclenche, CloudFormation procédez comme d'habitude à l'élimination des anciennes ressources. Si vous définissez une durée de surveillance mais que vous ne spécifiez aucun déclencheur d'annulation, CloudFormation attendez toujours la période spécifiée avant de nettoyer les anciennes ressources pour les opérations de mise à jour. Vous pouvez utiliser cette période de surveillance pour effectuer toute validation manuelle de pile souhaitée et annuler manuellement la création de la pile ou sa mise à jour, si nécessaire. Si vous définissez une surveillance de 0 minute, CloudFormation continue de surveiller les déclencheurs de restauration pendant les opérations de création et de mise à jour de la pile, et restaure l'opération su une alarme passe à l'état ALARM. Ensuite, pour

les opérations de mise à jour sans alarme de violation, la suppression des anciennes ressources commence immédiatement après que l'opération est terminée.

Par défaut, les opérations de pile CloudFormation ne sont annulées que si une alarme passe à ALARM l'état, et non à INSUFFICIENT_DATA l'état. Pour CloudFormation annuler le fonctionnement de la pile si une alarme passe également à INSUFFICIENT_DATA l'état, modifiez l' CloudWatch alarme pour traiter les données manquantes comme breaching. Pour plus d'informations, consultez la [section Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes](#) dans le guide de CloudWatch l'utilisateur Amazon.

CloudFormation ne surveille pas les déclencheurs de restauration lorsqu'il annule une pile au cours d'une opération de mise à jour.

Vous pouvez ajouter un maximum de 5 déclencheurs de restauration. Pour ajouter un déclencheur d'annulation, spécifiez l'Amazon Resource Name (ARN) de l' CloudWatch alarme. Actuellement, les types `AWS::CloudWatch::Alarm` et `AWS::CloudWatch::CompositeAlarm` peuvent être utilisés comme déclencheurs de restauration. Pour plus d'informations sur les CloudWatch alarmes, consultez la section [Utilisation des CloudWatch alarmes](#) dans le guide de CloudWatch l'utilisateur Amazon.

Si une CloudWatch alarme donnée est absente, l'ensemble de l'opération de pile échoue et revient en arrière.

Sachez que l'accès à CloudWatch nécessite des informations d'identification. Ces informations d'identification doivent être autorisées à accéder aux AWS ressources, par exemple pour récupérer des données CloudWatch métriques relatives à vos ressources. Pour plus d'informations, consultez [Authentification et contrôle d'accès CloudWatch](#) dans le guide de CloudWatch l'utilisateur Amazon.

Ajouter des déclencheurs de restauration lors de la création ou de la mise à jour d'une pile

Pour ajouter des déclencheurs de restauration lors de la création ou de la mise à jour d'une pile (console)

1. Lors de la création ou de la mise à jour d'une pile, sur la page Configurer les options de la pile, sous Options avancées, développez la section Configuration de restauration.
2. Saisissez une durée de surveillance comprise entre 0 et 180 minutes. La valeur par défaut est 0.
3. Spécifiez l'ARN de l' CloudWatch alarme ou de l'alarme composite que vous souhaitez utiliser comme déclencheur d'annulation, puis choisissez Ajouter un ARN CloudWatch d'alarme.

Par exemple, voici un ARN pour une CloudWatch alarme ou une alarme composite, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Sélectionnez Suivant et vérifiez les détails de votre pile.
5. Lorsque vous êtes prêt, sélectionnez Soumettre pour créer ou mettre à jour la pile.

Pour ajouter des déclencheurs de restauration lors de la création ou de la mise à jour d'une pile (AWS CLI)

Utilisez la commande [create-stack](#) ou [update-stack](#) avec l'option `--rollback-configuration`.

Par exemple, la `update-stack` commande suivante est définie `MyCompositeAlarm` comme déclencheur de restauration avec une période de surveillance de 5 minutes :

```
aws cloudformation update-stack --stack-name MyStack \  
  --use-previous-template \  
  --rollback-configuration \  
  "RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-  
east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTim
```

Ajouter des déclencheurs de restauration à un jeu de modifications

Pour ajouter des déclencheurs de restauration à un jeu de modifications (console)

1. Lors de la création d'un jeu de modifications, sur la page Configurer les options de pile, sous Options avancées, développez la section Configuration de restauration.
2. Saisissez une durée de surveillance comprise entre 0 et 180 minutes. La valeur par défaut est 5.
3. Spécifiez l'ARN de l' CloudWatch alarme ou de l'alarme composite que vous souhaitez utiliser comme déclencheur d'annulation, puis choisissez Ajouter un ARN CloudWatch d'alarme.

Par exemple, voici un ARN pour une CloudWatch alarme ou une alarme composite, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Sélectionnez Suivant et vérifiez les détails de votre jeu de modifications.
5. Lorsque vous êtes prêt, sélectionnez Créer un jeu de modifications pour créer l'ensemble de modifications.

Pour ajouter des déclencheurs de restauration à un ensemble de modifications (AWS CLI)

Utilisez la commande [create-change-set](#) avec l'option `--rollback-configuration`.

Afficher les déclencheurs de restauration pour une pile

Pour afficher les déclencheurs de restauration d'une pile, veuillez consulter la section Configuration de la restauration.

1. Sur la page Piles, choisissez la pile que vous voulez afficher dans la liste de gauche.
2. Dans l'onglet Informations sur la pile, développez la section Configuration de restauration pour afficher les déclencheurs de restauration.

Afficher les déclencheurs de restauration pour un jeu de modifications

Pour afficher les déclencheurs de restauration d'un jeu de modifications, veuillez consulter la section Configuration de la restauration.

1. Sur la page Piles, choisissez la pile que vous voulez afficher dans la liste de gauche.
2. Sélectionnez l'onglet Jeux de modifications, puis sélectionnez le jeu de modifications que vous voulez afficher.
3. Sélectionnez l'onglet Entrée et affichez la section Configuration de restauration.

Détectez les modifications de configuration non gérées apportées aux piles et aux ressources grâce à la détection des écarts

Même si vous gérez vos ressources via CloudFormation, les utilisateurs peuvent modifier ces ressources en dehors de CloudFormation. Les utilisateurs peuvent modifier une ressource directement en utilisant le service sous-jacent qui a créé la ressource. Par exemple, vous pouvez utiliser la console Amazon EC2 pour mettre à jour une instance de serveur créée dans le cadre d'une CloudFormation pile. Certaines modifications peuvent être accidentelles, et d'autres peuvent être apportées intentionnellement pour répondre aux événements opérationnels prioritaires. Quoi qu'il en soit, les modifications apportées en dehors de CloudFormation peuvent compliquer les opérations de mise à jour ou de suppression de la pile. Vous pouvez utiliser la détection de dérive pour identifier les ressources de pile auxquelles des modifications de configuration ont été apportées en dehors de la CloudFormation gestion. Vous pouvez alors prendre des mesures correctives pour resynchroniser les ressources de pile avec leurs définitions dans le modèle de pile. Vous pouvez, par exemple, mettre à jour directement les ressources ayant dérivé, pour les accorder avec leur définition de modèle. La

résolution de l'écart permet de garantir la cohérence de la configuration et la réussite des opérations de pile.

Rubriques

- [Qu'est-ce qu'un écart ?](#)
- [Codes de statut de détection de l'écart](#)
- [Considérations lors de la détection de l'écart](#)
- [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#)
- [Déterminer l'écart sur des ressources de pile individuelles](#)
- [Résoudre l'écart avec une opération d'importation](#)

Qu'est-ce qu'un écart ?

La détection de l'écart vous permet de détecter si la configuration réelle d'une pile diffère, ou a dérivé, de sa configuration prévue. CloudFormation À utiliser pour détecter la dérive sur l'ensemble d'une pile ou sur des ressources individuelles au sein de la pile. Une ressource est considérée comme ayant dérivé dès qu'une de ses valeurs de propriété réelles diffère des valeurs de propriété prévues. Cela inclut l'éventualité où la propriété ou la ressource aurait été supprimée. Une pile est considérée comme ayant dérivé si une ou plusieurs de ses ressources ont dérivé.

Pour déterminer si une ressource a dérivé, CloudFormation détermine les valeurs attendues des propriétés de ressource, telles que définies dans le modèle de pile, ainsi que toutes les valeurs spécifiées en tant que paramètres du modèle. CloudFormation compare ensuite ces valeurs attendues aux valeurs réelles de ces propriétés de ressources telles qu'elles existent actuellement dans la pile. Une ressource est considérée comme ayant dérivé si une ou plusieurs de ses propriétés ont été supprimées ou modifiées.

CloudFormation génère des informations détaillées sur chaque ressource de la pile qui a dérivé.

CloudFormation détecte la dérive sur les AWS ressources qui prennent en charge la détection de la dérive. Les ressources qui ne prennent pas en charge la détection de l'écart se voient attribuer le statut d'écart NOT_CHECKED. Pour obtenir la liste des AWS ressources qui prennent en charge la détection de la dérive, consultez [Prise en charge des types de ressources](#).

En outre, CloudFormation prend en charge la détection de la dérive sur les types de ressources privées provisionnables, c'est-à-dire dont le type de provisionnement est ouFULLY_MUTABLE.

IMMUTABLE Pour effectuer une détection de l'écart sur une ressource de type privé, la version par défaut du type de ressource que vous avez enregistré dans votre compte doit pouvoir être alloué. Pour plus d'informations sur le type de provisionnement des ressources, consultez le paramètre `ProvisioningType` de l'action [DescribeType](#) dans la Référence d'API AWS CloudFormation et de la commande [DescribeType](#) dans la Référence des commandes AWS CLI . Pour plus d'informations sur les ressources privées, consultez [Gestion des extensions avec le CloudFormation registre](#).

Vous pouvez exécuter la détection de l'écart sur des piles dotées des statuts suivants : `CREATE_COMPLETE`, `UPDATE_COMPLETE`, `UPDATE_ROLLBACK_COMPLETE` et `UPDATE_ROLLBACK_FAILED`.

Lors de la détection d'une dérive sur une pile, CloudFormation ne détecte aucune dérive sur les piles imbriquées appartenant à cette pile. Pour de plus amples informations, veuillez consulter [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#). Au lieu de cela, vous pouvez lancer une opération de détection de l'écart directement sur la pile imbriquée.

Note

CloudFormation détermine uniquement la dérive pour les valeurs de propriété définies de manière explicite, soit via le modèle de pile, soit en spécifiant les paramètres du modèle. Ceci n'inclut pas les valeurs par défaut des propriétés des ressources. Pour CloudFormation suivre une propriété de ressource afin de déterminer la dérive, définissez explicitement la valeur de la propriété, même si vous la définissez sur la valeur par défaut.

Codes de statut de détection de l'écart

Les tableaux de cette section décrivent les différents types de statut utilisés avec la détection de l'écart :

- Le statut d'opération de détection de l'écart décrit l'état actuel de l'opération.
- Statut de l'écart

Pour les ensembles de piles, cette information décrit le statut de l'écart de l'ensemble de pile dans son ensemble, en fonction du statut de l'écart des instances de piles qui lui appartiennent.

Pour les instances de piles, cette information décrit le statut de l'écart de l'instance de pile, en fonction du statut de l'écart de la pile associée.

Pour les piles, cette information décrit le statut de l'écart de la pile dans son ensemble, en fonction du statut de l'écart de ses ressources.

- Le statut de l'écart de ressource décrit le statut de l'écart d'une ressource individuelle.

Le tableau suivant répertorie les codes d'état CloudFormation attribués aux opérations de détection de la dérive des piles.

Statut de l'opération de détection de l'écart	Description
DETECTION_COMPLETE	L'opération de détection de l'écart d'une pile a réussi pour toutes les ressources de la pile qui prennent en charge la détection de l'écart.
DETECTION_FAILED	L'opération de détection de l'écart de la pile a échoué pour au moins une ressource dans la pile. Les résultats seront disponibles pour les ressources ayant effectué CloudFormation avec succès la détection de la dérive.
DETECTION_IN_PROGRESS	L'opération de détection de l'écart de pile est actuellement en cours.

Le tableau suivant répertorie les codes d'état de dérive CloudFormation attribués aux piles.

Statut de l'écart	Description
DRIFTED	Pour les piles : la pile diffère, ou présente un écart, par rapport à la configuration de modèle prévue. Une pile est considérée comme ayant dérivé si une ou plusieurs de ses ressources ont dérivé. Pour les instances de piles : une instance de pile est considérée comme présentant un écart si la pile présente un écart.

Statut de l'écart	Description
	Pour les ensembles de piles : un ensemble de piles est considéré comme présentant un écart si une ou plusieurs instances de piles présente un écart.
NOT_CHECKED	CloudFormation n'a pas vérifié si la pile, l'ensemble de piles ou l'instance de pile diffère de la configuration de modèle attendue.
IN_SYNC	La configuration actuelle de chaque ressource prise en charge correspond à sa configuration de modèle prévue. Le statut d'une pile, d'un ensemble de piles ou d'une instance de pile sans ressource prenant en charge la détection de l'écart sera également IN_SYNC.

Le tableau suivant répertorie les codes d'état de dérive CloudFormation attribués aux ressources de la pile.

Statut de l'écart de ressource	Description
DELETED	La ressource diffère de sa configuration de modèle prévue, car la ressource a été supprimée.
MODIFIED	La ressource diffère de sa configuration de modèle prévue.
NOT_CHECKED	CloudFormation n'a pas vérifié si la ressource diffère de la configuration de modèle attendue.
IN_SYNC	La configuration actuelle de la ressource correspond à sa configuration de modèle prévue.

Le tableau suivant répertorie les codes d'état de type de différence CloudFormation attribués aux propriétés des ressources qui diffèrent de leur configuration de modèle attendue.

Types de différences de propriété	Description
ADD	Une valeur a été ajoutée à une propriété de ressource qui est un type de données de tableau ou de liste.
REMOVE	La propriété a été supprimée de la configuration de ressource actuelle.
NOT_EQUAL	La valeur de propriété actuelle diffère de sa valeur prévue, définie dans le modèle de pile.

Considérations lors de la détection de l'écart

Afin d'effectuer correctement la détection de l'écart sur une pile, un utilisateur doit disposer des autorisations suivantes :

- L'autorisation de lecture pour chaque ressource de la pile qui prend en charge la détection de l'écart. Par exemple, si la pile comprend une ressource `AWS::EC2::Instance`, vous devez avoir l'autorisation `ec2:DescribeInstances` nécessaire pour effectuer la détection de l'écart sur la pile.
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`
- `cloudformation:BatchDescribeTypeConfigurations`

Pour plus d'informations sur la définition des autorisations dans CloudFormation, consultez [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

Dans certains cas extrêmes, il CloudFormation se peut que vous ne puissiez pas toujours renvoyer des résultats de dérive précis. Vous devez être conscient de ces cas limites afin d'interpréter correctement vos résultats de détection de l'écart.

- Dans certains cas, les objets contenus dans des tableaux de propriétés sont signalés comme un écart, alors qu'ils sont en réalité des valeurs par défaut fournies à la propriété à partir du service sous-jacent responsable de la ressource.

- Certaines ressources ont des relations d'attachement avec des ressources connexes, de telle sorte qu'une ressource peut en fait attacher ou supprimer les valeurs de propriété d'une autre ressource, définie dans le même modèle ou dans un autre modèle. Par exemple, les ressources `AWS::EC2::SecurityGroupIngress` et `AWS::EC2::SecurityGroupEgress` peuvent être utilisées pour attacher et supprimer des valeurs à partir des ressources `AWS::EC2::SecurityGroup`. Dans ces cas, CloudFormation analyse le modèle de pile pour détecter les pièces jointes avant d'effectuer la comparaison de dérive. Cependant, CloudFormation vous ne pouvez pas effectuer cette analyse sur plusieurs piles et vous risquez donc de ne pas renvoyer de résultats de dérive précis lorsque les ressources associées résident dans des piles différentes.

Les ressources qui prennent en charge la détection de l'écart et autorisent ou nécessitent des attachements à partir d'autres ressources incluent :

Type de ressource	Type de ressource d'attachement
<code>AWS::SNS::Topic</code>	<code>AWS::SNS::Subscription</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::UserToGroupAddition</code>
<code>AWS::IAM::Group</code>	<code>AWS::IAM::ManagedPolicy</code>
<code>AWS::IAM::Role</code>	<code>AWS::IAM::Policy</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::RolePolicy</code>
<code>AWS::ElasticLoadBalancingV2::Listener</code>	<code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code>
<code>AWS::EC2::SecurityGroup</code>	<code>AWS::EC2::SecurityGroupEgress</code> <code>AWS::EC2::SecurityGroupIngress</code>

- CloudFormation n'effectue aucune détection de dérive sur la `KMSKeyId` propriété d'aucune ressource. Comme AWS KMS les clés peuvent être référencées par plusieurs alias, nous ne CloudFormation pouvons pas garantir des résultats de dérive toujours précis pour cette propriété.
- Certaines propriétés de ressources que vous pouvez spécifier dans votre modèle de pile ne CloudFormation pourront pas, de par leur nature même, être comparées aux propriétés des

ressources de pile résultantes. Ces propriétés ne peuvent donc pas être incluses dans les résultats de détection de l'écart. Ces propriétés se divisent en deux grandes catégories :

- Valeurs de propriété qui CloudFormation ne peuvent pas être mappées à leur valeur de propriété de ressource initiale dans le modèle de pile.

Par exemple, il est CloudFormation impossible de mapper le code source d'une fonction Lambda au type de `Code` propriété de la `AWS::Lambda::Function` ressource et ne CloudFormation peut donc pas l'inclure dans les résultats de détection de dérive.

- Valeurs de propriété que le service responsable de la ressource ne renvoie pas.

Il existe certaines valeurs de propriété qui, par conception, ne sont jamais renvoyées par le service auquel la ressource appartient. Elles ont tendance à contenir des informations confidentielles, telles que des mots de passe ou d'autres données sensibles qui ne doivent pas être exposées. Par exemple, le service IAM ne renverra jamais la valeur de la `Password` propriété du type de `AWS::IAM::User LoginProfile` propriété et ne CloudFormation peut donc pas l'inclure dans les résultats de détection de dérive.

- Les objets dans un tableau peuvent être réellement des paramètres par défaut du service, et non pas d'un écart ajouté manuellement.
- Si vous recevez un faux positif, envoyez-nous vos commentaires en utilisant le lien de commentaires dans la CloudFormation console, ou contactez-nous via [AWS re:Post](#).
- Certaines propriétés peuvent avoir des valeurs d'entrée égales, mais non identiques. Pour éviter les faux positifs, vous devez vous assurer que la configuration attendue correspond à la configuration réelle.
- Par exemple, la configuration attendue d'une ressource peut être de 1 024 Mo et la configuration réelle de cette même ressource peut être de 1 Go. 1 024 Mo et 1 Go sont égaux, mais pas identiques.

Lorsque la détection des écarts s'exécute sur cette ressource, elle signale les résultats présentant des écarts.

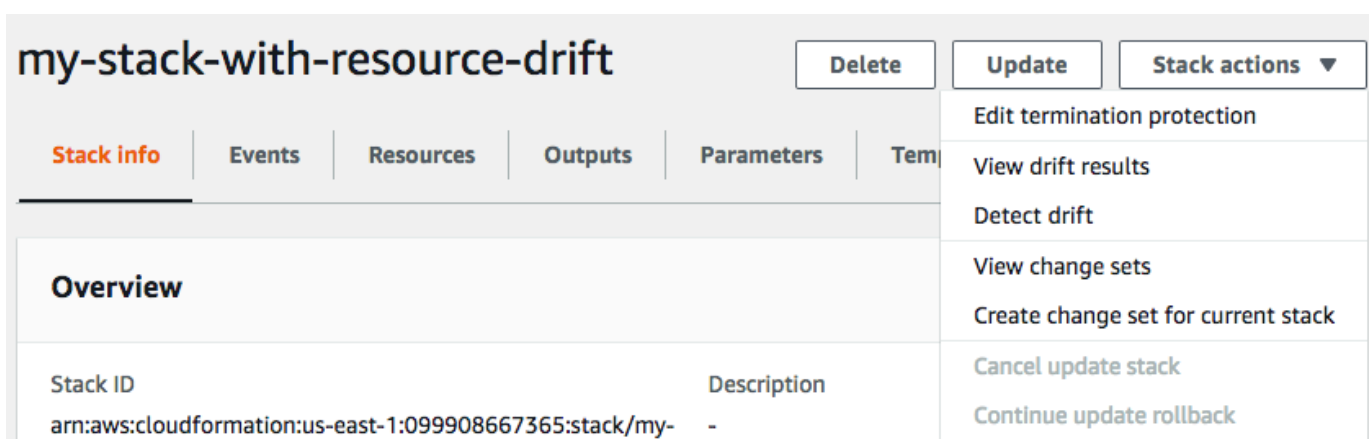
Pour éviter ce faux positif, changez la configuration attendue de la propriété de la ressource à 1 024 Mo, puis lancez la détection des écarts.

Détectez la dérive sur l'ensemble d'une CloudFormation pile

L'exécution d'une opération de détection de l'écart sur une pile détermine si la pile a dérivé de sa configuration de modèle prévue, et renvoie des informations détaillées sur le statut de l'écart de chaque ressource de la pile qui prend en charge la détection de l'écart.

Pour détecter la dérive sur l'ensemble d'une pile à l'aide du AWS Management Console

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la liste des piles, sélectionnez la pile sur laquelle vous souhaitez effectuer une détection de l'écart. Dans le volet des détails de la pile, choisissez Stack actions (Actions de pile), puis choisissez Detect drift (Détecter l'écart).



CloudFormation affiche une barre d'information indiquant que la détection de dérive a été initiée pour la pile sélectionnée.

3. Attendez que l'opération de détection de dérive soit CloudFormation terminée. Lorsque l'opération de détection de dérive est terminée, l'état de dérive et l'heure du dernier contrôle de dérive de votre pile sont mis à CloudFormation jour. Ces champs sont répertoriés dans la section Présentation du volet Stack info (Infos de pile) de la page des détails de la pile.

L'opération de détection de l'écart peut prendre plusieurs minutes, selon le nombre de ressources incluses dans la pile. Vous ne pouvez exécuter qu'une seule opération de détection de dérive sur une pile donnée à la fois. CloudFormation poursuit l'opération de détection de dérive même après avoir fermé la barre d'informations.

4. Passez en revue les résultats de détection de l'écart pour la pile et ses ressources. Votre pile étant sélectionnée, dans le menu Stack actions (Actions de pile), sélectionnez View drift results (Afficher les résultats d'écart).

CloudFormation répertorie l'état de dérive global de la pile, en plus de la dernière fois que la détection de dérive a été initiée sur la pile ou sur l'une de ses ressources individuelles. Une pile est considérée comme ayant dérivé si une ou plusieurs de ses ressources ont dérivé.

CloudFormation > Stacks: my-stack-with-resource-drift > Drifts

Drifts

[Refresh](#) [Detect stack drift](#)

Stack drift status
Drift detection enables you to detect whether a stack's actual configuration differs, or has drifted, from its template configuration. [Learn more.](#)

Drift status	Last drift check time
DRIFTED	Fri, 24 May 2019 16:08:12 GMT

Dans la section État de dérive des ressources, CloudFormation répertorie chaque ressource de pile, son état de dérive et la dernière fois que la détection de dérive a été initiée sur la ressource. L'ID logique et l'ID physique de chaque ressource sont affichés pour vous aider à l'identifier. En outre, pour les ressources dont le statut est MODIFIÉ, CloudFormation affiche les détails de la dérive des ressources.

Vous pouvez trier les ressources en fonction de leur statut d'écart à l'aide de la colonne Statut de l'écart.

- Pour afficher les détails sur une ressource modifiée.
 - La ressource modifiée étant sélectionnée, sélectionnez View drift details (Afficher les détails de l'écart).

CloudFormation affiche la page détaillée de dérive pour cette ressource. Cette page répertorie les valeurs de propriété prévue et actuelle de la ressource, ainsi que toutes les différences entre les deux.

Pour mettre en surbrillance une différence, sélectionnez le nom de la propriété dans la section Différences.

- Les propriétés ajoutées sont surlignées en vert dans la colonne Actuels de la section Détails.
- Les propriétés supprimées sont surlignées en rouge dans la colonne Expected (Prévus) de la section Détails.

- Les propriétés dont les valeurs ont été modifiées sont surlignées en jaune dans les deux colonnes Expected et Actuels.

Differences (4)

<1>

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName":"stuff","AttributeType":"S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key":"test","Value":"test"}]	-

Details

Expected

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Pour détecter la dérive sur l'ensemble d'une pile à l'aide du AWS CLI

Important

Vérifiez la valeur Heure du dernier contrôle de dérive pour la pile et confirmez qu'elle est antérieure à l'horodatage affiché dans les résultats de dérive de la ressource afin d'éviter l'utilisation de données obsolètes.

Pour détecter la dérive sur l'ensemble d'une pile à l'aide de AWS CLI, utilisez les AWS CLI commandes suivantes :

- `detect-stack-drift` pour lancer une opération de détection de l'écart sur une pile.
- `describe-stack-drift-detection-status` pour surveiller le statut de l'opération de détection de l'écart de la pile.
- `describe-stack-resource-drifts` pour passer en revue les détails de l'opération de détection de l'écart de la pile.

1. Utilisez `detect-stack-drift` pour détecter l'écart sur une pile complète. Spécifiez le nom ou l'ARN de la pile. Vous pouvez également spécifier la logique IDs des ressources spécifiques que vous souhaitez utiliser comme filtres pour cette opération de détection de dérive.

```
aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
```

Sortie :

```
{
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
}
```

2. Comme les opérations de détection d'écart de la pile peuvent être longues, utilisez `describe-stack-drift-detection-status` pour surveiller le statut de l'opération d'écart. Cette commande prend l'ID de détection de l'écart de pile renvoyé par la commande `detect-stack-drift`.

Dans l'exemple ci-dessous, nous avons pris l'ID de détection de l'écart de pile renvoyé par l'exemple `detect-stack-drift` ci-dessus et l'avons transmis comme paramètre à `describe-stack-drift-detection-status`. Le paramètre renvoie les détails de l'opération qui montrent que l'opération

de détection de l'écart s'est terminée, qu'une seule ressource de pile a dérivé et que la pile complète est considérée comme ayant dérivé, au final.

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

Sortie :

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample",
  "StackDriftStatus": "DRIFTED",
  "Timestamp": "2018-03-26T17:23:22.279Z",
  "DetectionStatus": "DETECTION_COMPLETE",
  "DriftedStackResourceCount": 1
}
```

- Une fois l'opération de détection de l'écart de pile terminée, utilisez la commande `describe-stack-resource-drifts` pour passer en revue les résultats, y compris les valeurs de propriété réelles et prévues des ressources ayant dérivé.

L'exemple ci-dessous utilise l'option `--stack-resource-drift-status-filters` permettant de demander des informations de dérive de pile uniquement pour les ressources qui ont été modifiées ou supprimées. La demande renvoie des informations sur la seule ressource qui a été modifiée, y compris des détails sur deux de ses propriétés dont les valeurs ont été modifiées. Aucune ressource n'a été supprimée.

```
aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-resource-drift --stack-resource-drift-status-filters MODIFIED DELETED
```

Sortie :

```
{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0, \"DelaySeconds\":120, \"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
```

```

    east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
    \"maxReceiveCount\":12},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize
    \":262144,\"VisibilityTimeout\":60,\"QueueName\": \"my-stack-with-resource-drift-
    Queue-494PBHC076H4\"}],
    \"ResourceType\": \"AWS::SQS::Queue\",
    \"Timestamp\": \"2018-03-26T17:23:34.489Z\",
    \"PhysicalResourceId\": \"https://sqs.us-
    east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-
    Queue-494PBHC076H4\",
    \"StackResourceDriftStatus\": \"MODIFIED\",
    \"ExpectedProperties\": \"{\\\"ReceiveMessageWaitTimeSeconds\\\":0,
    \\\"DelaySeconds\\\":20,\\\"RedrivePolicy\\\":{\\\"deadLetterTargetArn\\\":\\\"arn:aws:sqs:us-
    east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\\\",
    \\\"maxReceiveCount\\\":10},\\\"MessageRetentionPeriod\\\":345600,\"MaximumMessageSize
    \":262144,\"VisibilityTimeout\":60,\"QueueName\": \"my-stack-with-resource-drift-
    Queue-494PBHC076H4\"}]\",
    \"PropertyDifferences\": [
      {
        \"PropertyPath\": \"/DelaySeconds\",
        \"ActualValue\": \"120\",
        \"ExpectedValue\": \"20\",
        \"DifferenceType\": \"NOT_EQUAL\"
      },
      {
        \"PropertyPath\": \"/RedrivePolicy/maxReceiveCount\",
        \"ActualValue\": \"12\",
        \"ExpectedValue\": \"10\",
        \"DifferenceType\": \"NOT_EQUAL\"
      }
    ],
    \"LogicalResourceId\": \"Queue\"
  }
]
}

```

Détecter l'écart sur des ressources de pile individuelles

Vous pouvez détecter l'écart sur des ressources spécifiques au sein d'une pile, plutôt que sur la pile complète. Cela est particulièrement utile lorsque vous avez uniquement besoin de déterminer si des ressources spécifiques correspondent à nouveau à leurs configurations de modèle prévues.

Lorsque vous effectuez une détection de dérive sur une ressource, met CloudFormation également à jour l'état général de dérive de la pile et l'heure du dernier contrôle de dérive, le cas échéant. Par exemple, supposons qu'une pile ait un état de dérive `IN_SYNC`. Vous devez CloudFormation effectuer une détection de dérive sur une ou plusieurs ressources contenues dans cette pile et CloudFormation détecter qu'une ou plusieurs de ces ressources ont dérivé. CloudFormation met à jour l'état de dérive de la pile sur `DRIFTED`. Inversement, supposons qu'une pile ait un état de dérive `DRIFTED` en raison d'une seule ressource dérivée. Si vous redéfinissez les valeurs de propriété attendues pour cette ressource, puis que vous détectez à nouveau une dérive sur la ressource, l'état de dérive de la ressource et le statut de dérive de la pile CloudFormation seront mis à jour `IN_SYNC` sans que vous ayez à détecter à nouveau la dérive sur l'ensemble de la pile.

Pour détecter la dérive d'une ressource individuelle à l'aide du AWS Management Console

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la liste des piles, sélectionnez la pile contenant la ressource. CloudFormation affiche les détails de cette pile.
3. Dans le volet de navigation de gauche, sous Piles, choisissez Actions de pile, puis Détecter l'écart.
4. Sous Statut de l'écart de ressource, choisissez la ressource, puis sélectionnez Détecter l'écart pour une ressource.

CloudFormation effectue une détection de dérive sur la ressource sélectionnée. En cas de succès, CloudFormation met à jour l'état de dérive de la ressource et l'état de dérive global de la pile, si nécessaire. CloudFormation met également à jour l'horodatage indiquant la date à laquelle la détection de dérive a été effectuée pour la dernière fois sur la ressource et sur la pile dans son ensemble. Si la ressource a été modifiée, CloudFormation affiche des informations de dérive détaillées sur les valeurs de propriété attendues et actuelles de la ressource.

5. Passez en revue les résultats de détection de l'écart pour la ressource.
 - Pour afficher les détails sur une ressource modifiée.
 - La ressource modifiée étant sélectionnée, sélectionnez View drift details (Afficher les détails de l'écart).

CloudFormation affiche les détails de la dérive pour cette ressource, y compris les valeurs de propriété attendues et actuelles de la ressource, ainsi que les différences entre les deux.

- Pour mettre en surbrillance une différence, sélectionnez le nom de la propriété dans la section Différences.
- Les propriétés ajoutées sont surlignées en vert dans la colonne Actuels de la section Détails.
 - Les propriétés supprimées sont surlignées en rouge dans la colonne Expected (Prévus) de la section Détails.
 - Les propriétés dont les valeurs ont été modifiées sont surlignées en jaune dans les deux colonnes Expected et Actuels.

Differences (4)

< 1 >

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName":"stuff","AttributeType":"S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key":"test","Value":"test"}]	-

Details

Expected

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Pour détecter la dérive d'une ressource individuelle à l'aide du AWS CLI

- **⚠ Important**

Vérifiez la valeur Heure du dernier contrôle de dérive pour la ressource de la pile et confirmez qu'elle est antérieure à l'horodatage affiché dans les résultats de dérive de la ressource afin d'éviter l'utilisation de données obsolètes.

Pour détecter une dérive sur une ressource individuelle à l'aide de AWS CLI, utilisez la `detect-stack-resource-drift` commande. Spécifiez l'ID logique de la ressource et la pile dans laquelle elle se trouve.

L'exemple suivant exécute une opération de détection de dérive sur une ressource spécifique de la pile, `my-drifted-resource`. La réponse renvoie des informations qui confirment que la ressource a été modifiée, y compris des détails sur deux de ses propriétés dont les valeurs ont été modifiées.

```
aws cloudformation detect-stack-resource-drift \
  --stack-name my-stack-with-resource-drift \
  --logical-resource-id my-drifted-resource
```

Sortie :

```
{
  "StackResourceDrift": {
    "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
    "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds\":120,\"RedrivePolicy\":{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\"},\"maxReceiveCount\":12},\"MessageRetentionPeriod\":345600,\"MaximumMessageSize\":262144,\"VisibilityTimeout\":60,\"QueueName\":\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}\",
    "ResourceType": "AWS::SQS::Queue",
    "Timestamp": "2018-03-26T18:54:28.462Z",
    "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-Queue-494PBHC076H4",
    "StackResourceDriftStatus": "MODIFIED",
```

```

    "ExpectedProperties": "{ \"ReceiveMessageWaitTimeSeconds\":0,
    \"DelaySeconds\":20, \"RedrivePolicy\":{ \"deadLetterTargetArn\": \"arn:aws:sqs:us-
    east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
    \"maxReceiveCount\":10}, \"MessageRetentionPeriod\":345600, \"MaximumMessageSize
    \":262144, \"VisibilityTimeout\":60, \"QueueName\": \"my-stack-with-resource-drift-
    Queue-494PBHC076H4\" }",
    "PropertyDifferences": [
      {
        "PropertyPath": "/DelaySeconds",
        "ActualValue": "120",
        "ExpectedValue": "20",
        "DifferenceType": "NOT_EQUAL"
      },
      {
        "PropertyPath": "/RedrivePolicy/maxReceiveCount",
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
      }
    ],
    "LogicalResourceId": "my-drifted-resource"
  }
}

```

Résoudre l'écart avec une opération d'importation

Dans certains cas, la configuration d'une ressource peut s'écarter de sa configuration prévue ; vous pouvez alors souhaiter accepter la nouvelle configuration comme configuration prévue. Dans la plupart des cas, vous pouvez résoudre les résultats de l'écart en mettant à jour la définition de ressource dans le modèle de pile avec une nouvelle configuration, puis effectuer une mise à jour de pile. Toutefois, si la nouvelle configuration met à jour une propriété de ressource qui nécessite un remplacement, la ressource sera recrée lors de la mise à jour de la pile. Si vous souhaitez conserver la ressource existante, vous pouvez utiliser la fonction d'importation de ressource pour mettre à jour la ressource et résoudre les résultats de l'écart sans provoquer le remplacement de la ressource.

La résolution de l'écart d'une ressource par le biais d'une opération d'importation se compose des étapes de base suivantes :

- [Ajoutez un DeletionPolicy attribut, défini sur Conserver, à la ressource](#). Cela garantit que la ressource existante est conservée et non supprimée lorsqu'elle est retirée de la pile.
- [Retirez la ressource du modèle et exécutez une opération de mise à jour de la pile](#). La ressource est retirée de la pile, mais n'est pas supprimée.
- [Décrivez l'état réel de la ressource dans le modèle de pile, puis réimportez la ressource existante dans la pile](#). La ressource est à nouveau ajoutée dans la pile et les différences de propriété qui causaient les résultats de l'écart sont résolues.

Pour plus d'informations sur l'importation de ressources, voir [Importation manuelle de AWS ressources dans une CloudFormation pile](#). Pour obtenir la liste des ressources qui prennent en charge l'importation, consultez [Prise en charge des types de ressources](#).

Dans cet exemple, nous utilisons le modèle suivant, nommé `templateToImport.json`.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

```

    }
  },
  "GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "Properties": {
      "TableName": "Games",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "BillingMode": "PROVISIONED",
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH

```

```

BillingMode: PROVISIONED
ProvisionedThroughput:
  ReadCapacityUnits: 5
  WriteCapacityUnits: 1
GamesTable:
  Type: AWS::DynamoDB::Table
  Properties:
    TableName: Games
    AttributeDefinitions:
      - AttributeName: key
        AttributeType: S
    KeySchema:
      - AttributeName: key
        KeyType: HASH
    BillingMode: PROVISIONED
    ProvisionedThroughput:
      ReadCapacityUnits: 5
      WriteCapacityUnits: 1

```

Dans cet exemple, supposons qu'un utilisateur a modifié une ressource en dehors de CloudFormation. Après avoir exécuté la détection de l'écart, nous avons découvert que GamesTable a été modifié BillingMode en PAY_PER_REQUEST. Pour plus d'informations sur la détection d'écart, voir [Déterminez les modifications de configuration non gérées apportées aux piles et aux ressources grâce à la détection des écarts](#).

Differences (5)					< 1 >
☒	Property	Change	Expected value	Current value	
☒	BillingMode	NOT_EQUAL	PROVISIONED	PAY_PER_REQUEST	
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	0	
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	1	0	
Details					
Expected			Actual		
<pre> { "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PROVISIONED", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 5, "WriteCapacityUnits": 1 }, "TableName": "Games" } </pre>			<pre> { "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PAY_PER_REQUEST", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 0, "WriteCapacityUnits": 0 }, "TableName": "Games" } </pre>		

Notre pile est maintenant obsolète, nos ressources sont actives, mais nous voulons préserver la configuration de ressources prévue. Pour ce faire, nous pouvons résoudre l'écart via une opération d'importation, sans interrompre les services.

Résolvez la dérive avec une opération d'importation à l'aide de la CloudFormation console

Étape 1. Mettre à jour la pile avec la politique de suppression Retain

Pour mettre à jour la pile à l'aide d'un attribut **DeletionPolicy** avec l'option **Retain**

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Stacks (Piles) choisissez la pile qui a fait l'objet de l'écart.
3. Choisissez Update (Mettre à jour), puis Replace current template (Remplacer le modèle actuel) dans le volet des détails de la pile.
4. Dans la page Specify template (Spécifier un modèle), indiquez votre modèle mis à jour contenant l'attribut DeletionPolicy avec l'option Retain via une des méthodes suivantes :
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.

Ensuite, choisissez Suivant.

5. Vérifiez la page Specify stack details (Spécifier les détails de la pile) et choisissez Next (Suivant).
6. Vérifiez la page Configure stack options (Configurer les options de la pile) et choisissez Next (Suivant).
7. Sur la *stack-name* page de révision, choisissez Update stack.

Résultats : Sur la page Events (Événements) de votre pile, le statut est UPDATE_COMPLETE.

Pour résoudre le problème de dérive lors d'une opération d'importation, sans interrompre les services, spécifiez a Retain [DeletionPolicy](#) pour les ressources que vous souhaitez supprimer de votre pile. Dans l'exemple suivant, nous avons ajouté un [DeletionPolicy](#) attribut, défini sur Retain, à la GameTable ressource.

Exemple JSON

```
"GameTable": {
```



```
"Type": "AWS::DynamoDB::Table",  
"DeletionPolicy": "Retain",  
"Properties": {  
    "TableName": "Games",
```

Example YAML

```
GamesTable:  
  Type: AWS::DynamoDB::Table  
  DeletionPolicy: Retain  
  Properties:  
    TableName: Games
```

Étape 2. Retirer les ressources présentant des écarts, les paramètres associés et les sorties

Pour retirer les ressources présentant des écarts, les paramètres associés et les sorties

1. Choisissez Update (Mettre à jour), puis Replace current template (Remplacer le modèle actuel) dans le volet des détails de la pile.
2. Dans la page Specify template (Spécifier un modèle), indiquez à votre modèle mis à jour ses ressources, ses paramètres associés et ses sorties supprimées du modèle de pile à l'aide de l'une des méthodes suivantes :
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.

Ensuite, choisissez Suivant.

3. Vérifiez la page Specify stack details (Spécifier les détails de la pile) et choisissez Next (Suivant).
4. Vérifiez la page Configure stack options (Configurer les options de la pile) et choisissez Next (Suivant).
5. Sur la *stack-name* page de révision, choisissez Update stack.

Résultats : L'ID logique GamesTable a un statut DELETE_SKIPPED sur la page Events (Événements) de votre pile.

Attendez que l'opération de mise à jour de la pile soit CloudFormation terminée. Une fois l'opération de mise à jour de la pile terminée, supprimez la ressource, les paramètres associés et les sorties du modèle de pile. Ensuite, importez le modèle mis à jour. Lorsque ces actions ont été effectuées, l'exemple de modèle ressemble à ce qui suit.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

Exemple YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
```

```

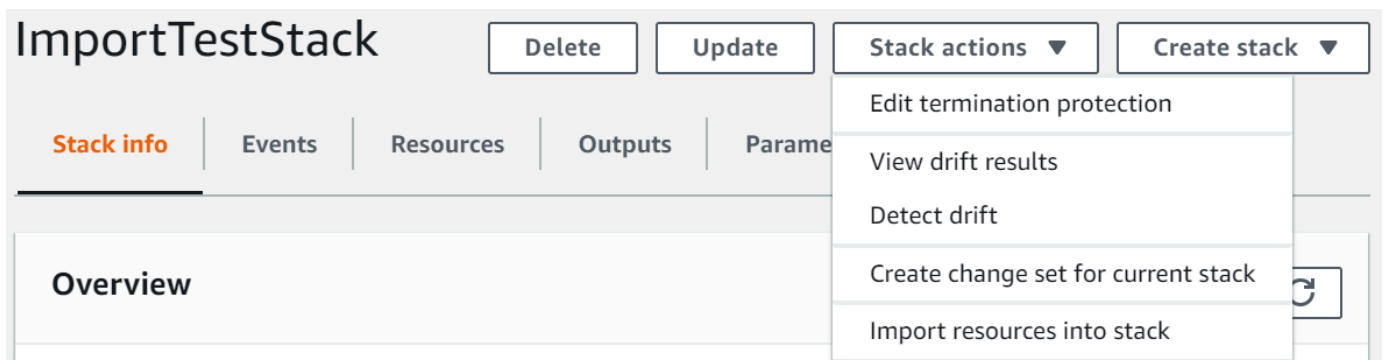
Type: AWS::DynamoDB::Table
Properties:
  TableName: Service
  AttributeDefinitions:
    - AttributeName: key
      AttributeType: S
  KeySchema:
    - AttributeName: key
      KeyType: HASH
  BillingMode: PROVISIONED
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 1

```

Étape 3. Mettre à jour le modèle afin qu'il corresponde à l'état réel de vos ressources

Pour mettre à jour le modèle afin qu'il corresponde à l'état réel des ressources

1. Pour importer le modèle mis à jour, choisissez **Stack actions** (Actions de la pile) puis **Import resources into stack** (Importer des ressources dans la pile).



2. Consultez la page **Import overview** (Vue d'ensemble de l'importation) pour obtenir la liste des éléments que vous devez fournir dans le cadre de cette opération, puis choisissez **Next** (Suivant).
3. Dans la page **Specify template** (Spécifier le modèle) indiquez votre modèle mis à jour via une des méthodes suivantes :
 - Choisissez **Amazon S3 URL** (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez **Upload a template file** (Charger un fichier de modèle), puis recherchez votre modèle.

Ensuite, choisissez Suivant.

4. Dans la page Identify resources (Identifier les ressources), identifiez chaque ressource cible. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 - a. Sous Identifier property (Propriété de l'identifiant), choisissez le type d'identifiant de ressource. Par exemple, la propriété TableName identifie la ressource `AWS::DynamoDB::Table`.
 - b. Sous Identifier value (Valeur de l'identifiant), entrez la valeur réelle de la propriété. Dans l'exemple de modèle, TableName pour la ressource GamesTable est Games.
 - c. Choisissez Suivant.
5. Consultez la page Specify stack details (Spécifier les détails de la pile), puis choisissez Next (Suivant).
6. Dans la page Import overview (Aperçu de l'importation), vérifiez les ressources importées, puis choisissez Import resources (Importer les ressources). Le type de ressource `AWS::DynamoDB::Table` sera à nouveau importé dans votre pile.

Résultats: Dans cet exemple, nous avons résolu l'écart de ressource via une opération d'importation, sans interrompre les services. Vous pouvez vérifier la progression d'une action d'importation dans la CloudFormation console, dans l'onglet Événements. Les ressources importées auront un état `IMPORT_COMPLETE` suivi d'un statut `CREATE_COMPLETE` avec Resource import complete (Importation de ressource terminée) comme raison du statut.

Attendez que l'opération de mise à jour de la pile soit CloudFormation terminée. Une fois l'opération de mise à jour de la pile terminée, mettez à jour votre modèle de sorte qu'il corresponde à l'état de l'écart réel de vos ressources. Par exemple, BillingMode sera défini sur `PAY_PER_REQUEST`, et ReadCapacityUnits et WriteCapacityUnits seront définis sur 0.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
```

```
        "TableName": "Service",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    },
    "GamesTable": {
        "Type": "AWS::DynamoDB::Table",
        "DeletionPolicy": "Retain",
        "Properties": {
            "TableName": "Games",
            "AttributeDefinitions": [
                {
                    "AttributeName": "key",
                    "AttributeType": "S"
                }
            ],
            "KeySchema": [
                {
                    "AttributeName": "key",
                    "KeyType": "HASH"
                }
            ],
            "BillingMode": "PAY_PER_REQUEST",
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 0,
                "WriteCapacityUnits": 0
            }
        }
    }
}
```

```
}  
}
```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09  
Description: Import test  
Resources:  
  ServiceTable:  
    Type: AWS::DynamoDB::Table  
    Properties:  
      TableName: Service  
      AttributeDefinitions:  
        - AttributeName: key  
          AttributeType: S  
      KeySchema:  
        - AttributeName: key  
          KeyType: HASH  
      BillingMode: PROVISIONED  
      ProvisionedThroughput:  
        ReadCapacityUnits: 5  
        WriteCapacityUnits: 1  
  GamesTable:  
    Type: AWS::DynamoDB::Table  
    DeletionPolicy: Retain  
    Properties:  
      TableName: Games  
      AttributeDefinitions:  
        - AttributeName: key  
          AttributeType: S  
      KeySchema:  
        - AttributeName: key  
          KeyType: HASH  
      BillingMode: PAY_PER_REQUEST  
      ProvisionedThroughput:  
        ReadCapacityUnits: 0  
        WriteCapacityUnits: 0
```

Importer AWS des ressources dans une CloudFormation pile

Vous pouvez importer des ressources existantes dans une CloudFormation pile. Cela est utile si vous souhaitez commencer à utiliser CloudFormation pour gérer des ressources créées en dehors de CloudFormation, sans avoir à les supprimer ni à les recréer.

CloudFormation propose les options suivantes pour importer des ressources existantes dans une pile :

- [Le générateur IaC](#) est un outil qui analyse automatiquement vos ressources existantes et génère un CloudFormation modèle en fonction de leur état actuel. Ce modèle peut ensuite être utilisé pour importer ces ressources dans une pile.
- L'[importation de ressources](#) est un processus manuel dans lequel vous décrivez les ressources existantes dans votre CloudFormation modèle, puis vous les importez dans une pile. Cette approche vous oblige à spécifier manuellement les propriétés et les configurations des ressources dans le modèle.
- L'[importation automatique](#) est un processus automatique dans lequel vous décrivez les ressources existantes dans votre CloudFormation modèle et CloudFormation importe celles dont les noms personnalisés correspondent dans une pile.
- La [refactorisation des piles](#) est une fonctionnalité qui simplifie la réorganisation des ressources de vos CloudFormation piles tout en préservant les propriétés et les données des ressources existantes. Grâce à la refactorisation des piles, vous pouvez déplacer des ressources entre les piles, diviser des piles monolithiques en composants plus petits ou consolider plusieurs piles en une seule.

Outre la CloudFormation gestion des ressources existantes, la fonctionnalité d'importation de ressources peut être utile dans les scénarios suivants :

- Déplacement de ressources entre les piles : vous pouvez importer des ressources d'une pile vers une autre, ce qui vous permet de réorganiser votre infrastructure selon vos besoins.
- Imbrication de piles existantes : vous pouvez importer une pile existante en tant que pile imbriquée dans une autre pile, ce qui permet de concevoir des infrastructures modulaires et réutilisables.

CloudFormation prend en charge l'importation d'un large éventail de ressources. Pour de plus amples informations, veuillez consulter [Prise en charge des types de ressources](#).

Rubriques

- [Importation manuelle de AWS ressources dans une CloudFormation pile](#)
- [Importez automatiquement AWS des ressources dans une CloudFormation pile](#)
- [Rétablissement d'une opération d'importation](#)

Importation manuelle de AWS ressources dans une CloudFormation pile

Avec l'importation de ressources, vous pouvez importer AWS des ressources existantes dans une CloudFormation pile nouvelle ou existante. Au cours d'une opération d'importation, vous créez un jeu de modifications qui importe vos ressources existantes dans une pile ou crée une nouvelle pile à partir de vos ressources existantes. Vous fournissez les éléments suivants lors de l'importation.

- Un modèle qui décrit l'ensemble de la pile, y compris les ressources de la pile d'origine et les ressources que vous importez. Chaque ressource à importer doit avoir un [attribut DeletionPolicy](#).
- Identifiants des ressources que vous importez qui CloudFormation peuvent être utilisés pour mapper la logique IDs du modèle avec les ressources existantes.

Note

CloudFormation ne prend en charge qu'un seul niveau d'imbrication à l'aide de l'importation de ressources. Cela signifie que vous ne pouvez pas importer une pile dans une pile enfant ou importer une pile qui a des enfants.

Rubriques

- [Identifiants de ressource](#)
- [Validation de l'importation de ressources](#)
- [Codes d'état de l'importation des ressources](#)
- [Considérations à prendre en compte lors d'une opération d'importation](#)
- [Ressources supplémentaires](#)
- [Création d'une pile à partir de ressources existantes](#)
- [Importation de ressources existantes dans une pile](#)
- [Déplacement de ressources d'une pile à une autre](#)

- [Imbrication d'une pile existante](#)

Identifiants de ressource

Vous fournissez deux valeurs pour identifier chaque ressource que vous importez.

- Une propriété d'identifiant. Il s'agit d'une propriété de ressource qui peut être utilisée pour identifier chaque type de ressource. Par exemple, une ressource `AWS::S3::Bucket` peut être identifiée à l'aide de son code `BucketName`.

La propriété de ressource que vous utilisez pour identifier la ressource que vous importez varie en fonction du type de ressource. Vous pouvez trouver la propriété de la ressource dans la CloudFormation console. Après avoir créé un modèle incluant la ressource à importer, vous pouvez lancer le processus d'importation, où vous trouverez les propriétés d'identifiant des ressources que vous importez. Pour certains types de ressources, il peut exister plusieurs façons de les identifier, et vous pouvez sélectionner la propriété à utiliser dans les listes déroulantes.

Vous pouvez également obtenir les propriétés d'identification des ressources que vous importez en appelant la commande [get-template-summary](#) CLI et en spécifiant l'URL S3 du modèle de pile comme valeur de l'option `--template-url`.

- Une valeur d'identifiant. Il s'agit de la valeur immobilière réelle de la ressource. Par exemple, la valeur réelle de la propriété `BucketName` peut être `MyS3Bucket`.

Vous pouvez obtenir la valeur de la propriété d'identifiant à partir de la console de service pour la ressource.

Validation de l'importation de ressources

Au cours d'une opération d'importation, CloudFormation effectue les validations suivantes.

- La ressource à importer existe.
- Les propriétés et les valeurs de configuration de chaque ressource à importer sont conformes au schéma du type de ressource, qui définit ses propriétés acceptées, les propriétés requises et les valeurs de propriété prises en charge.
- Les propriétés requises sont spécifiées dans le modèle. Les propriétés requises pour chaque type de ressource sont décrites dans la [référence des types de ressources et de propriétés AWS](#).
- La ressource à importer n'appartient pas à une autre pile dans la même région.

CloudFormation ne vérifie pas que la configuration du modèle correspond à la configuration réelle des propriétés des ressources.

⚠ Important

Vérifiez que les ressources et leurs propriétés définies dans le modèle correspondent à la configuration prévue de l'importation de ressources afin d'éviter des modifications inattendues.

Codes d'état de l'importation des ressources

Ce tableau décrit les différents types de statut utilisés avec la fonctionnalité d'importation de ressources.

État de l'opération d'importation	Description
IMPORT_IN_PROGRESS	L'opération d'importation est en cours.
IMPORT_COMPLETE	L'opération d'importation a été exécutée pour toutes les ressources de la pile.
IMPORT_ROLLBACK_IN_PROGRESS	L'opération de restauration de l'importation restaure la configuration précédente du modèle.
IMPORT_ROLLBACK_FAILED	L'opération de restauration de l'importation a échoué.
IMPORT_ROLLBACK_COMPLETE	L'importation est restaurée à la configuration précédente du modèle.

Considérations à prendre en compte lors d'une opération d'importation

- Une fois l'importation terminée et avant d'effectuer les opérations de pile suivantes, nous vous conseillons d'exécuter la détection de l'écart sur les ressources importées. La détection de l'écart garantit que la configuration du modèle correspond à la configuration réelle. Pour de plus amples informations, veuillez consulter [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#).
- Les opérations d'importation n'autorisent pas les créations de ressources, les suppressions de ressources ou les modifications apportées aux configurations de propriétés.

- Chaque ressource à importer doit avoir un attribut `DeletionPolicy` pour que l'opération d'importation réussisse. L'attribut `DeletionPolicy` peut être défini sur n'importe quelle valeur possible. Seules les ressources que vous importez ont besoin d'un `DeletionPolicy`. Les ressources qui font déjà partie de la pile n'ont pas besoin d'un attribut `DeletionPolicy`.
- Vous ne pouvez pas importer la même ressource dans plusieurs piles.
- Vous pouvez utiliser la condition de politique IAM `cloudformation:ImportResourceTypes` pour contrôler les types de ressources que les utilisateurs peuvent utiliser au cours d'une opération d'importation. Pour de plus amples informations, veuillez consulter [Clés de conditions de politique pour CloudFormation](#).
- Les limites de CloudFormation pile s'appliquent lors de l'importation de ressources. Pour plus d'informations sur les limites, consultez [Comprendre les CloudFormation quotas](#).

Ressources supplémentaires

Pour résoudre le problème de l'écart de pile avec une importation de ressources, voir [Résoudre l'écart avec une opération d'importation](#).

Création d'une pile à partir de ressources existantes

Cette rubrique explique comment créer une pile à partir de AWS ressources existantes en les décrivant dans un modèle. Pour rechercher plutôt les ressources existantes et générer automatiquement un modèle que vous pouvez utiliser pour importer des ressources existantes dans un nouveau compte CloudFormation ou les répliquer dans un nouveau compte, voir [Générez des modèles à partir de ressources existantes avec le générateur laC](#).

Conditions préalables

Avant de commencer, les prérequis suivants doivent être remplis :

- Un modèle qui décrit toutes les ressources qui doivent faire partie de la nouvelle pile. Enregistrez le modèle localement ou dans un compartiment Amazon S3.
- Pour chaque ressource que vous souhaitez importer, incluez les éléments suivants :
 - les propriétés et les valeurs de propriétés qui définissent la configuration actuelle de la ressource.
 - identifiant unique de la ressource, tel que le nom de la ressource. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 - [l'attribut DeletionPolicy](#).

Rubriques

- [Exemple de modèle](#)
- [Créez une pile à partir de ressources existantes à l'aide du AWS Management Console](#)
- [Créez une pile à partir de ressources existantes à l'aide du AWS CLI](#)

Exemple de modèle

Dans cette procédure pas à pas, nous supposons que vous utilisez l'exemple de modèle suivant, appelé `TemplateToImport.json`, qui spécifie deux tables DynamoDB créées en dehors de. CloudFormation `ServiceTable` et `GameTable` sont les cibles de l'importation.

Note

Ce modèle n'est fourni qu'à titre d'exemple. Pour l'utiliser à vos propres fins de test, remplacez les exemples de ressources par des ressources provenant de votre compte.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
```

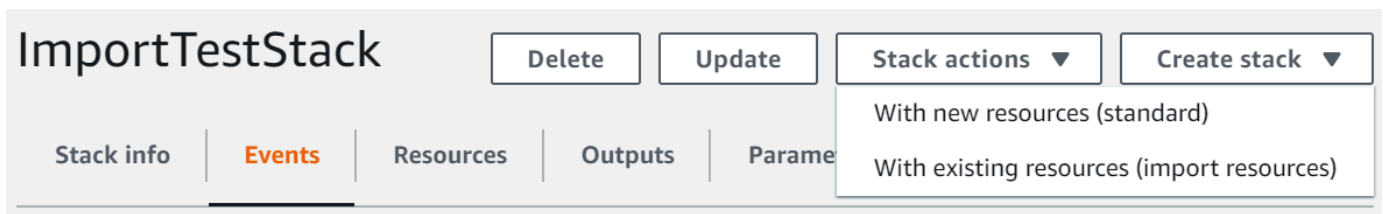
```

        "WriteCapacityUnits": 1
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}

```

Créez une pile à partir de ressources existantes à l'aide du AWS Management Console

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la page Piles choisissez Créer une pile, puis choisissez With existing resources (import resources) (Avec des ressources existantes (importer des ressources)).



3. Lisez la page Vue d'ensemble de l'importation pour obtenir la liste des éléments que vous devez fournir dans le cadre de cette opération. Ensuite, choisissez Suivant.
4. Dans la page Specify template (Spécifier un modèle) indiquez votre modèle à l'aide de l'une des méthodes suivantes, puis choisissez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.
5. Dans la page Identify resources (Identifier les ressources), identifiez chaque ressource cible. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 - a. Sous Identifier property (Propriété de l'identifiant), choisissez le type d'identifiant de ressource. Par exemple, la ressource `AWS::DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`.
 - b. Sous Identifier value (Valeur de l'identifiant), tapez la valeur réelle de la propriété. Par exemple, la valeur de `TableName` pour la ressource `GamesTable` dans l'exemple de modèle est *Games*.
 - c. Choisissez Suivant.
6. Dans la page Specify stack details (Spécifier les détails de la pile), modifiez les paramètres souhaités, puis choisissez Suivant. Un jeu de modifications est créé automatiquement.

 Important

L'opération d'importation échoue si vous modifiez des paramètres existants qui laissent une opération de création, de mise à jour ou de suppression.

7. Sur la *stack-name* page Révision, vérifiez que les bonnes ressources sont importées, puis choisissez Importer des ressources. Le jeu de modifications créé à la dernière étape est alors automatiquement exécuté.

Le panneau Événements de la page Stack details (Détails de la pile) pour votre nouvelle pile s'affiche.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

8. (Facultatif) Exécutez la détection de l'écart sur la pile afin de vous assurer que le modèle et la configuration réelle des ressources importées correspondent. Pour plus d'informations sur la détection de l'écart, voir [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#).
9. (Facultatif) Si vos ressources importées ne correspondent pas aux configurations de modèle attendues, corrigez les configurations de modèle ou mettez directement à jour les ressources. Dans cette démonstration, nous corrigeons les configurations des modèles afin qu'elles correspondent à leurs configurations réelles.
 - a. [Rétablissez l'opération d'importation](#) des ressources affectées.
 - b. Ajoutez à nouveau les cibles d'importation à votre modèle, en vous assurant que les configurations de modèle correspondent aux configurations réelles.
 - c. Répétez les étapes 2 à 8 à l'aide du modèle modifié afin d'importer à nouveau les ressources.

Créez une pile à partir de ressources existantes à l'aide du AWS CLI

1. Pour savoir quelles propriétés identifient chaque type de ressource dans le modèle, exécutez la commande `get-template-summary` en spécifiant l'URL S3 du modèle. Par exemple, la ressource `AWS::DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`. Pour la ressource `GameTable` de l'exemple de modèle, la valeur de `TableName` est `Games`. Vous aurez besoin de ces informations à l'étape suivante.

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).

2. Composez une liste des ressources réelles de votre modèle et de leurs identifiants uniques dans le format de chaîne JSON suivant.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}, {"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":{"TableName":"Service"}}]
```

Sinon, vous pouvez spécifier les paramètres au format JSON dans un fichier de configuration.

Par exemple, pour importer ServiceTable et GamesTable, vous pouvez créer un *ResourcesToImport.txt* fichier contenant la configuration suivante.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier":{
      "TableName":"Games"
    }
  },
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"ServiceTable",
    "ResourceIdentifier":{
      "TableName":"Service"
    }
  }
]
```

3. Pour créer un ensemble de modifications, utilisez la commande create-change-set suivante et remplacez le texte de l'espace réservé. Pour l'option --change-set-type, indiquez une valeur de **IMPORT**. Pour l'option --resources-to-import, remplacez l'exemple de chaîne JSON par la chaîne JSON que vous venez de créer.

```
aws cloudformation create-change-set \
  --stack-name TargetStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
```



```
--template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/  
TemplateToImport.json \  
--resources-to-  
import '["ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifi  
{"TableName":"Games"}},  
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifi  
{"TableName":"Service"}]']
```

Note

--resources-to-import ne prend pas en charge le YAML en ligne. Les exigences relatives aux guillemets d'échappement dans la chaîne JSON varient en fonction de votre terminal. Pour plus d'informations, consultez la section [Utilisation des guillemets dans les chaînes de caractères](#) du Guide de l'utilisateur AWS Command Line Interface .

Vous pouvez également utiliser l'URL d'un fichier en entrée pour l'option --resources-to-import, comme indiqué dans l'exemple suivant.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Vérifiez le jeu de modifications pour vous assurer que les ressources appropriées seront importées.

```
aws cloudformation describe-change-set \  
--change-set-name ImportChangeSet --stack-name TargetStack
```

5. Pour lancer le jeu de modifications et importer les ressources, utilisez la commande execute-change-set suivante et remplacez le texte de remplacement. Une fois l'opération (IMPORT_COMPLETE) terminée, les ressources sont importées avec succès.

```
aws cloudformation execute-change-set \  
--change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Facultatif) Exécutez la détection de l'écart sur la pile IMPORT_COMPLETE pour vous assurer que le modèle et la configuration réelle des ressources importées correspondent. Pour plus d'informations sur la détection de l'écart, voir [Détection de l'écart sur des ressources de pile individuelles](#).

a. Exécutez la détection d'écart sur la pile spécifiée.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Si elle aboutit, cette commande renvoie un résultat similaire à ce qui suit.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Afficher la progression d'une opération de détection d'écart pour l'ID de détection d'écart de pile spécifié.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Affichez les informations d'écart pour les ressources dont l'écart a été vérifiée dans la pile spécifiée.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Facultatif) Si vos ressources importées ne correspondent pas aux configurations de modèle attendues, corrigez les configurations de modèle ou mettez directement à jour les ressources. Dans cette démonstration, nous corrigeons les configurations des modèles afin qu'elles correspondent à leurs configurations réelles.
 - a. [Rétablissez l'opération d'importation](#) des ressources affectées.
 - b. Ajoutez à nouveau les cibles d'importation à votre modèle, en vous assurant que les configurations de modèle correspondent aux configurations réelles.
 - c. Répétez les étapes 3 à 6 en utilisant le modèle modifié pour réimporter les ressources.

Importation de ressources existantes dans une pile

Cette rubrique explique comment importer AWS des ressources existantes dans une pile existante en les décrivant dans un modèle. Pour rechercher plutôt les ressources existantes et générer automatiquement un modèle que vous pouvez utiliser pour importer des ressources existantes dans un nouveau compte CloudFormation ou les répliquer dans un nouveau compte, voir [Générez des modèles à partir de ressources existantes avec le générateur laC](#).

Conditions préalables

Avant de commencer, les prérequis suivants doivent être remplis :

- Un modèle qui décrit l'ensemble de la pile, y compris les ressources qui font déjà partie de la pile et les ressources à importer. Enregistrez le modèle localement ou dans un compartiment Amazon S3.

Pour obtenir une copie du modèle d'une pile en cours d'exécution

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation/>.
 2. Dans la liste des piles, choisissez la pile dont vous souhaitez récupérer le modèle.
 3. Dans le volet des détails de la pile, choisissez l'onglet Modèle, puis choisissez Copier dans le presse-papiers.
 4. Collez le code dans un éditeur de texte pour commencer à ajouter d'autres ressources au modèle.
- Pour chaque ressource que vous souhaitez importer, incluez les éléments suivants :
 - les propriétés et les valeurs de propriétés qui définissent la configuration actuelle de la ressource.
 - identifiant unique de la ressource, tel que le nom de la ressource. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 - [l'attribut DeletionPolicy](#).

Rubriques

- [Exemple de modèle](#)
- [Importez une ressource existante dans une pile à l'aide du AWS Management Console](#)
- [Importez une ressource existante dans une pile à l'aide du AWS CLI](#)

Exemple de modèle

Dans cette procédure pas à pas, nous partons du principe que vous utilisez le modèle d'exemple suivant, appelé `TemplateToImport.json`, qui spécifie deux tables DynamoDB. `ServiceTable` fait actuellement partie de la pile et `GameTable` est la table que vous souhaitez importer.

Note

Ce modèle n'est fourni qu'à titre d'exemple. Pour l'utiliser à vos propres fins de test, remplacez les exemples de ressources par des ressources provenant de votre compte.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}
```

```

        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}

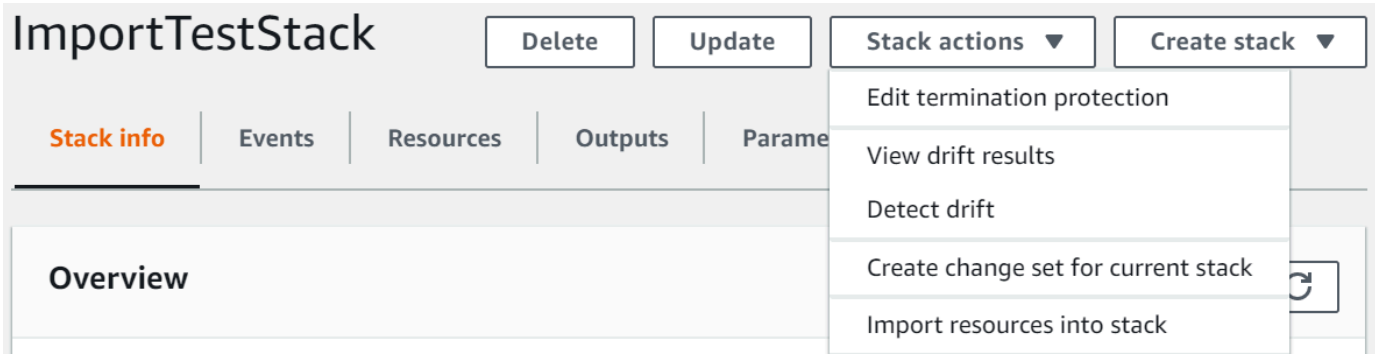
```

Importez une ressource existante dans une pile à l'aide du AWS Management Console

Note

La CloudFormation console ne prend pas en charge l'utilisation de la fonction intrinsèque `Fn::Transform` lors de l'importation de ressources. Vous pouvez utiliser le AWS CLI pour importer les ressources qui utilisent cette `Fn::Transform` fonction.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la page Piles choisissez la pile dans laquelle vous souhaitez importer des ressources.
3. Choisissez Stack actions (Actions de pile), puis Import resources into stack (Importer les ressources dans la pile).



4. Vérifiez la page Import overview (Aperçu de l'importation) puis choisissez Suivant.
5. Dans la page Specify template (Spécifier le modèle) indiquez votre modèle mis à jour à l'aide de l'une des méthodes suivantes, puis choisissez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.

- Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.
6. Dans la page Identify resources (Identifier les ressources), identifiez chaque ressource cible. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 - a. Sous Identifier property (Propriété de l'identifiant), choisissez le type d'identifiant de ressource. Par exemple, la ressource `AWS::DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`.
 - b. Sous Identifier value (Valeur de l'identifiant), tapez la valeur réelle de la propriété. Par exemple, la valeur de `TableName` pour la ressource `GameTable` dans l'exemple de modèle est *Games*.
 - c. Choisissez Suivant.
 7. Dans la page Specify stack details (Spécifier les détails de la pile), mettez à jour les paramètres, puis choisissez Suivant. Un jeu de modifications est créé automatiquement.

 Note

L'opération d'importation échoue si vous modifiez des paramètres existants qui laissent une opération de création, de mise à jour ou de suppression.

8. Sur la *stack-name* page Révision, passez en revue les ressources à importer, puis choisissez Importer des ressources. Le jeu de modifications créé à la dernière étape est alors automatiquement exécuté. Toutes les balises au niveau de la pile sont appliquées aux ressources importées à ce moment-là. Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#).

La page Événements de la pile s'affiche.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SamplelamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SamplelamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SamplelamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SamplelamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

- (Facultatif) Exécutez la détection de l'écart sur la pile afin de vous assurer que le modèle et la configuration réelle des ressources importées correspondent. Pour plus d'informations sur la détection de l'écart, voir [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#).
- (Facultatif) Si vos ressources importées ne correspondent pas aux configurations de modèle attendues, corrigez les configurations de modèle ou mettez directement à jour les ressources. Pour plus d'informations sur l'importation de ressources dérivées, voir [Résoudre l'écart avec une opération d'importation](#).

Importez une ressource existante dans une pile à l'aide du AWS CLI

- Pour savoir quelles propriétés identifient chaque type de ressource dans le modèle, exécutez la commande `get-template-summary` en spécifiant l'URL S3 du modèle. Par exemple, la ressource `AWS::DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`. Pour la ressource `GameTable` de l'exemple de modèle, la valeur de `TableName` est `Games`. Vous aurez besoin de ces informations à l'étape suivante.

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).

- Composez une liste des ressources réelles à importer et de leurs identifiants uniques dans le format de chaîne JSON suivant.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

Sinon, vous pouvez spécifier les paramètres au format JSON dans un fichier de configuration.

Par exemple, pour importer `GamesTable`, vous pouvez créer un *ResourcesToImport.txt* fichier contenant la configuration suivante.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier": {
      "TableName":"Games"
    }
  }
]
```

3. Pour créer un ensemble de modifications, utilisez la commande `create-change-set` suivante et remplacez le texte de l'espace réservé. Pour l'option `--change-set-type`, indiquez une valeur de **IMPORT**. Pour l'option `--resources-to-import`, remplacez l'exemple de chaîne JSON par la chaîne JSON que vous venez de créer.

```
aws cloudformation create-change-set \
  --stack-name TargetStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/  
TemplateToImport.json \
  --resources-to-import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]'
```

Note

`--resources-to-import` ne prend pas en charge le YAML en ligne. Les exigences relatives aux guillemets d'échappement dans la chaîne JSON varient en fonction de votre terminal. Pour plus d'informations, consultez la section [Utilisation des guillemets dans les chaînes de caractères](#) du Guide de l'utilisateur AWS Command Line Interface.

Vous pouvez également utiliser l'URL d'un fichier en entrée pour l'option `--resources-to-import`, comme indiqué dans l'exemple suivant.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Vérifiez le jeu de modifications pour vous assurer que les ressources appropriées seront importées.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Pour lancer le jeu de modifications et importer les ressources, utilisez la commande `execute-change-set` suivante et remplacez le texte de remplacement. Toutes les balises au niveau de la pile sont appliquées aux ressources importées à ce moment-là. Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#). Une fois l'opération (IMPORT_COMPLETE) terminée, les ressources sont importées avec succès.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Facultatif) Exécutez la détection de l'écart sur la pile IMPORT_COMPLETE pour vous assurer que le modèle et la configuration réelle des ressources importées correspondent. Pour plus d'informations sur la détection de l'écart, voir [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#).

- a. Exécutez la détection d'écart sur la pile spécifiée.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Si elle aboutit, cette commande renvoie un résultat similaire à ce qui suit.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Afficher la progression d'une opération de détection d'écart pour l'ID de détection d'écart de pile spécifié.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Affichez les informations d'écart pour les ressources dont l'écart a été vérifiée dans la pile spécifiée.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Facultatif) Si vos ressources importées ne correspondent pas aux configurations de modèle attendues, corrigez les configurations de modèle ou mettez directement à jour les ressources. Pour plus d'informations sur l'importation de ressources dérivées, voir [Résoudre l'écart avec une opération d'importation](#).

Déplacement de ressources d'une pile à une autre

À l'aide de la fonction `resource import`, vous pouvez déplacer des ressources d'une pile à une autre ou les remanier, . Vous devez d'abord ajouter une politique de suppression Retain à la ressource que vous souhaitez déplacer pour vous assurer que la ressource est préservée lorsque vous la supprimez de la pile source et que vous l'importez dans la pile cible.

Si vous découvrez l'importation, nous vous recommandons dans un premier temps de passer en revue les informations d'introduction de la rubrique [Importer AWS des ressources dans une CloudFormation pile](#).

Important

Toutes les ressources ne prennent pas en charge les opérations d'importation. Veuillez consulter [Ressources prenant en charge les opérations d'importation](#) avant de supprimer une ressource de votre pile. Si vous supprimez de votre pile une ressource qui ne prend pas en charge les opérations d'importation, vous ne pouvez pas importer la ressource dans une autre pile ou la ramener dans la pile source.

Refactorisez une pile à l'aide du AWS Management Console

1. Dans le modèle de source, spécifiez un Retain [DeletionPolicy](#) pour la ressource que vous souhaitez déplacer.

Dans l'exemple de modèle source suivant, Games est la cible de ce remaniement.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}
```

```
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}
```

2. Ouvrez la CloudFormation console pour effectuer une mise à jour de la pile afin d'appliquer la politique de suppression.
 - a. Dans la page Piles, avec la pile sélectionnée, choisissez Mettre à jour.
 - b. Sous Prepare template (Préparer le modèle), choisissez Replace current template (Remplacer le modèle actuel).
 - c. Dans Specify template (Spécifier le modèle), indiquez le modèle source mis à jour avec l'attribut DeletionPolicy défini sur GameTable, puis choisissez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL du modèle source mis à jour dans la zone de texte.
 - Choisissez Upload a template file (Télécharger un fichier de modèle), puis recherchez le fichier de modèle source mis à jour.
 - d. Dans la page Specify stack details (Spécifier les détails de la pile) aucune modification n'est requise. Choisissez Suivant.
 - e. Dans la page Configure stack options (Configurer les options de pile) aucune modification n'est requise. Choisissez Suivant.
 - f. Sur la *SourceStackName* page Révision, passez en revue vos modifications. Si votre modèle contient des ressources IAM, sélectionnez I acknowledge that this template may create IAM resources (Je sais que ce modèle peut créer des ressources IAM) pour confirmer que vous souhaitez utiliser des ressources IAM dans le modèle. Pour plus d'informations sur l'utilisation de ressources IAM dans les modèles, consultez [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#). Mettez ensuite à jour votre pile source en créant un jeu de modifications ou mettez à jour votre pile source directement.
3. Supprimez la ressource, les paramètres associés et les sorties du modèle source, puis ajoutez-les au modèle cible.

Le modèle source ressemble maintenant à ce qui suit.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

L'exemple de modèle cible suivant a actuellement la ressource `PlayersTable` et contient maintenant également `GamesTable`.

Exemple JSON

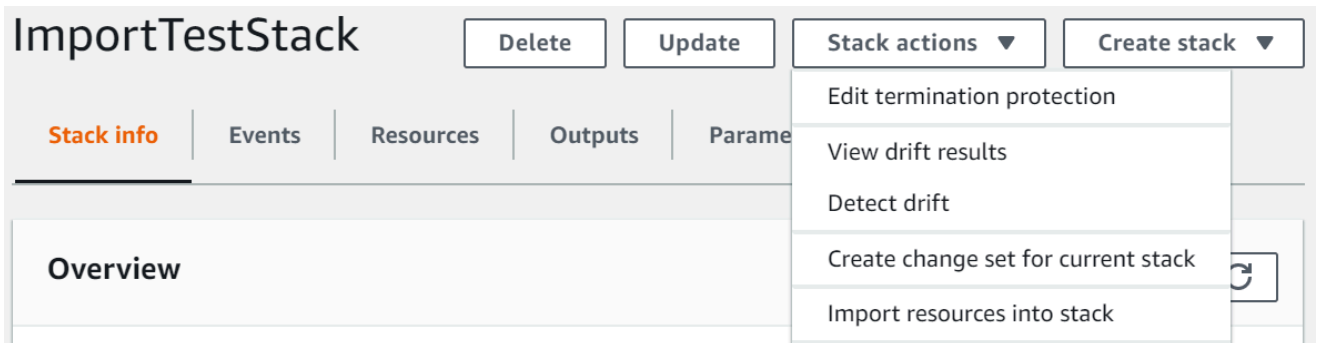
```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
```

```
"PlayersTable": {
  "Type": "AWS::DynamoDB::Table",
  "Properties": {
    "TableName": "Players",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
},
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

```

    }
  }
}
```

4. Répétez les étapes 2 et 3 pour mettre à jour à nouveau la pile source, afin, cette fois, de supprimer la ressource cible de la pile.
5. Effectuez une opération d'importation pour ajouter `GameTable` à la pile cible.
 - a. Sur la page Piles, sélectionnez la pile parent, choisissez Stack actions (Actions pour la pile), puis Import resources into stack (Importer les ressources dans la pile).



- b. Lisez la page Vue d'ensemble de l'importation pour obtenir la liste des éléments que vous devez fournir dans le cadre de cette opération. Ensuite, choisissez Suivant.
- c. Dans la page Specify template (Spécifier un modèle) effectuez l'une des opérations suivantes, puis choisissez Next (Suivant).
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez une URL dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez un fichier à télécharger.
- d. Dans la page Identify resources (Identifier les ressources) identifiez la ressource que vous déplacez (dans cet exemple, `GameTable`). Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).
 1. Sous Identifier property (Propriété de l'identifiant), choisissez le type d'identifiant de ressource. Par exemple, une ressource `AWS::DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`.
 2. Sous Identifier value (Valeur de l'identifiant), tapez la valeur réelle de la propriété. Par exemple, *GameTables*.
 3. Choisissez Suivant.

- e. Dans la page Specify stack details (Spécifier les détails de la pile), modifiez les paramètres souhaités, puis choisissez Suivant. Un jeu de modifications est créé automatiquement.

Important

L'opération d'importation échoue si vous modifiez des paramètres existants qui laissent une opération de création, de mise à jour ou de suppression.

- f. Sur la **TargetStackName** page Révision, vérifiez que la bonne ressource est importée, puis choisissez Importer des ressources. Le jeu de modifications lance la dernière étape est alors automatiquement exécuté. Toutes les [balises au niveau de la pile](#) sont appliquées aux ressources importées à ce moment-là.
- g. Le panneau Événements de la page Stack details (Détails de la pile) pour votre pile parent s'affiche.

ImportTestStack Delete Update Stack actions ▼ Create stack ▼

Stack info **Events** Resources Outputs Parameters Template Change sets

Events 🔄

🔍 Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	✔ IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIAMUser	✔ UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIAMUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIAMUser	✔ IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIAMUser	🔄 IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	🔄 IMPORT_IN_PROGRESS	User initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	🔄 REVIEW_IN_PROGRESS	User initiated

Note

Il n'est pas nécessaire d'exécuter la détection de l'écart sur la pile parent après cette opération d'importation, car la ressource `AWS::CloudFormation::Stack` est déjà gérée par CloudFormation.

Refactorisez une pile à l'aide du AWS CLI

1. Dans le modèle de source, spécifiez un Retain [DeletionPolicy](#) pour la ressource que vous souhaitez déplacer.

Dans l'exemple de modèle source suivant, `GameTable` est la cible de ce remaniement.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GameTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
```

```

        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}
}

```

2. Mettez à jour la pile source afin d'appliquer la politique de suppression à la ressource.

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. Supprimez la ressource, les paramètres associés, et les sorties du modèle source, puis ajoutez-les au modèle cible.

Le modèle source ressemble maintenant à ce qui suit.

Exemple JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}

```

```

    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}

```

L'exemple de modèle cible suivant a actuellement la ressource `PlayersTable` et contient maintenant également `GamesTable`.

Exemple JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "PlayersTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Players",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",

```

```

        "DeletionPolicy": "Retain",
        "Properties": {
            "TableName": "Games",
            "AttributeDefinitions": [
                {
                    "AttributeName": "key",
                    "AttributeType": "S"
                }
            ],
            "KeySchema": [
                {
                    "AttributeName": "key",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 1
            }
        }
    }
}

```

4. Mettez à jour la pile source afin de supprimer la ressource GamesTable et ses paramètres et sorties associés de la pile.

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. Composez une liste des ressources réelles à importer et de leurs identifiants uniques dans le format de chaîne JSON suivant. Pour de plus amples informations, veuillez consulter [Identifiants de ressource](#).

```
[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": "GamesTable", "PhysicalResourceId": "GamesTable", "ImportedResource": {"TableName": "Games"}}]
```

Sinon, vous pouvez spécifier les paramètres au format JSON dans un fichier de configuration.

Par exemple, pour importer GamesTable, vous pouvez créer un *ResourcesToImport.txt* fichier contenant la configuration suivante.

```
[
```

```
{
  "ResourceType": "AWS::DynamoDB::Table",
  "LogicalResourceId": "GamesTable",
  "ResourceIdentifier": {
    "TableName": "Games"
  }
}
```

6. Pour créer un ensemble de modifications, utilisez la commande `create-change-set` suivante et remplacez le texte de l'espace réservé. Pour l'option `--change-set-type`, indiquez une valeur de **IMPORT**. Pour l'option `--resources-to-import`, remplacez l'exemple de chaîne JSON par la chaîne JSON que vous venez de créer.

```
aws cloudformation create-change-set \
  --stack-name TargetStackName --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-import
  "'[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": {"TableName": "Games"}}]'"
```

Note

`--resources-to-import` ne prend pas en charge le YAML en ligne. Les exigences relatives aux guillemets d'échappement dans la chaîne JSON varient en fonction de votre terminal. Pour plus d'informations, consultez la section [Utilisation des guillemets dans les chaînes de caractères](#) du Guide de l'utilisateur AWS Command Line Interface .

Vous pouvez également utiliser l'URL d'un fichier en entrée pour l'option `--resources-to-import`, comme indiqué dans l'exemple suivant.

```
--resources-to-import file://ResourcesToImport.txt
```

7. Vérifiez le jeu de modifications pour vous assurer que la ressource correcte est importée dans la pile cible.

```
aws cloudformation describe-change-set \
```

```
--change-set-name ImportChangeSet
```

8. Pour lancer le jeu de modifications et importer la ressource, utilisez la commande `execute-change-set` suivante et remplacez le texte de remplacement. Toutes les balises au niveau de la pile sont appliquées aux ressources importées à ce moment-là. Une fois l'opération (IMPORT_COMPLETE) terminée, la ressource est importée avec succès.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStackName
```

Note

Il n'est pas nécessaire d'exécuter la détection de l'écart sur la pile cible après cette opération d'importation car la ressource est déjà gérée par CloudFormation.

Imbrication d'une pile existante

Utilisez la fonction `resource import` pour imbriquer une pile existante dans une autre pile existante. Les piles imbriquées sont des composants courants que vous déclarez et référencez à partir d'autres modèles. Vous pouvez ainsi éviter de copier et coller les mêmes configurations dans vos modèles et simplifier les mises à jour des piles. Si vous disposez d'un modèle pour un composant commun, vous pouvez utiliser la ressource `AWS::CloudFormation::Stack` pour référencer ce modèle à partir d'un autre modèle. Pour de plus amples informations sur les piles imbriquées, veuillez consulter [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).

CloudFormation ne prend en charge qu'un seul niveau d'imbrication en utilisant `resource import`. Cela signifie que vous ne pouvez pas importer une pile dans une pile enfant ou importer une pile qui a des enfants.

Si vous découvrez l'importation, nous vous recommandons dans un premier temps de passer en revue les informations d'introduction de la rubrique [Importation manuelle de AWS ressources dans une CloudFormation pile](#).

Validation de l'importation d'une pile imbriquée

Lors d'une opération d'importation de piles imbriquées, CloudFormation effectue les validations suivantes.

- La définition `AWS::CloudFormation::Stack` imbriquée dans le modèle de pile parent correspond au modèle de la pile imbriquée réelle.
- Les balises de la définition `AWS::CloudFormation::Stack` imbriquée dans le modèle de pile parent correspondent aux balises de la ressource imbriquée réelle.

Imbriguez une pile existante à l'aide du AWS Management Console

1. Ajoutez la `AWS::CloudFormation::Stack` ressource au modèle de pile parent avec un Retain [DeletionPolicy](#). Dans l'exemple de modèle de pile parent suivant, `MyNestedStack` est la cible de l'importation.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "KeySchema":[
          {
            "AttributeName":"key",
            "KeyType":"HASH"
          }
        ],
        "ProvisionedThroughput":{
          "ReadCapacityUnits":5,
          "WriteCapacityUnits":1
        }
      }
    },
    "MyNestedStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "DeletionPolicy": "Retain",
```

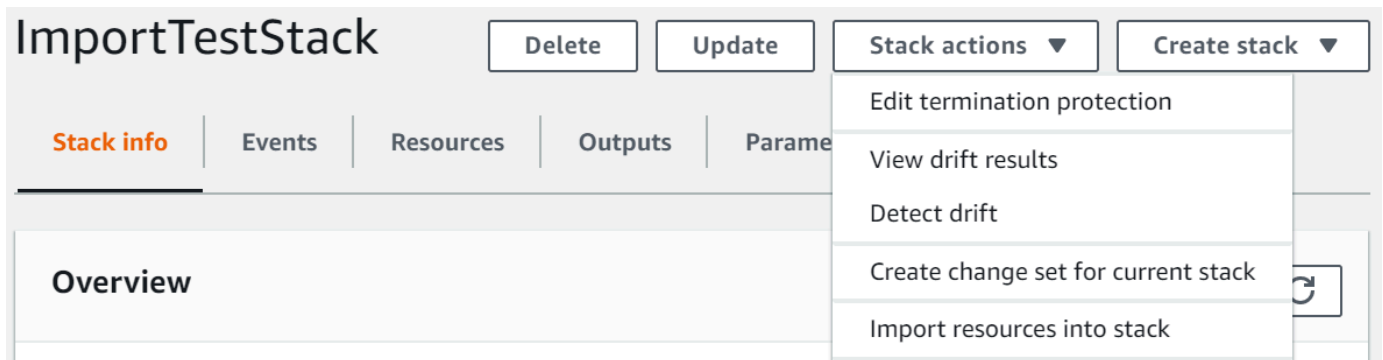
```
"Properties" : {
  "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
  "Parameters" : {
    "InstanceType" : "t1.micro",
    "KeyName" : "mykey"
  }
}
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  MyNestedStack:
    Type: AWS::CloudFormation::Stack
    DeletionPolicy: Retain
    Properties:
      TemplateURL: >-
        https://s3.amazonaws.com/cloudformation-templates-us-east-2/
        EC2ChooseAMI.template
      Parameters:
        InstanceType: t1.micro
        KeyName: mykey
```

2. Ouvrez la CloudFormation console.

3. Sur la page Piles, sélectionnez la pile parent, choisissez Stack actions (Actions pour la pile), puis Import resources into stack (Importer les ressources dans la pile).



4. Lisez la page Vue d'ensemble de l'importation pour obtenir la liste des éléments que vous devez fournir dans le cadre de cette opération. Ensuite, choisissez Suivant.
5. Dans la page Specify template (Spécifier le modèle) indiquez le modèle parent mis à jour à l'aide de l'une des méthodes suivantes, puis choisissez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.
6. Dans la page Identify resources (Identifier les ressources) identifiez la ressource `AWS::CloudFormation::Stack`.
 - a. Sous Identifier property (Propriété de l'identifiant), choisissez le type d'identifiant de ressource. Par exemple, une ressource `AWS::CloudFormation::Stack` peut être identifiée à l'aide de la propriété `StackId`.
 - b. Sous Valeur de l'identifiant, tapez l'ARN de la pile que vous importez. Par exemple, `arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10`.

Resources to import (1)

The following logical IDs are new to your template. Provide the identifier values for the resource you are importing.

NestedStack

AWS::CloudFormation::Stack

Identifier property

StackId

Identifier value

Enter StackId

c. Choisissez Suivant.

7. Dans la page Specify stack details (Spécifier les détails de la pile), modifiez les paramètres souhaités, puis choisissez Suivant. Un jeu de modifications est créé automatiquement.

Important

L'opération d'importation échoue si vous modifiez des paramètres existants qui laissent une opération de création, de mise à jour ou de suppression.

8. Sur la *MyParentStack* page Révision, vérifiez que la bonne ressource est importée, puis choisissez Importer des ressources. Le jeu de modifications créé à la dernière étape est alors automatiquement exécuté. Toutes les balises au niveau de la pile sont appliquées aux ressources importées à ce moment-là.
9. Le panneau Événements de la page Stack details (Détails de la pile) pour votre pile parent s'affiche.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

Note

Il n'est pas nécessaire d'exécuter la détection de l'écart sur la pile parent après cette opération d'importation, car la ressource `AWS::CloudFormation::Stack` a déjà été gérée par CloudFormation.

Imbriguez une pile existante à l'aide du AWS CLI

1. Ajoutez la `AWS::CloudFormation::Stack` ressource au modèle de pile parent avec un Retain [DeletionPolicy](#). Dans l'exemple de modèle parent suivant, `MyNestedStack` est la cible de l'importation.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ]
      }
    }
  }
}
```

```

        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    },
    "MyNestedStack" : {
        "Type" : "AWS::CloudFormation::Stack",
        "DeletionPolicy": "Retain",
        "Properties" : {
            "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
            "Parameters" : {
                "InstanceType" : "t1.micro",
                "KeyName" : "mykey"
            }
        }
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:

```

```

    ReadCapacityUnits: 5
    WriteCapacityUnits: 1
    MyNestedStack:
      Type: AWS::CloudFormation::Stack
      DeletionPolicy: Retain
      Properties:
        TemplateURL: >-
          https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template
        Parameters:
          InstanceType: t1.micro
          KeyName: mykey

```

2. Composez une chaîne JSON comme indiqué dans l'exemple suivant, avec les modifications suivantes :

- Remplacez *MyNestedStack* par l'ID logique de la ressource cible tel que spécifié dans le modèle.
- *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10* Remplacez-le par l'ARN de la pile que vous souhaitez importer.

```

[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","Resource
{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}]}]

```

Vous pouvez également spécifier les paramètres dans un fichier de configuration.

Par exemple, pour importer *MyNestedStack*, vous pouvez créer un *ResourcesToImport.txt* fichier contenant la configuration suivante.

JSON

```

[
  {
    "ResourceType":"AWS::CloudFormation::Stack",
    "LogicalResourceId":"MyNestedStack",
    "ResourceIdentifier": {
      "StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/
mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"
    }
  }
]

```

```

    }
  }
]
```

YAML

```

ResourceType: AWS::CloudFormation::Stack
LogicalResourceId: MyNestedStack
ResourceIdentifier:
  StackId: >-
    arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10
```

3. Pour créer un ensemble de modifications, utilisez la commande `create-change-set` suivante et remplacez le texte de l'espace réservé. Pour l'option `--change-set-type`, indiquez une valeur de **IMPORT**. Pour l'option `--resources-to-import`, remplacez l'exemple de chaîne JSON par la chaîne JSON que vous venez de créer.

```

aws cloudformation create-change-set \
  --stack-name MyParentStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-
import '[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack",
{"StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]'
```

Note

`--resources-to-import` ne prend pas en charge le YAML en ligne. Les exigences relatives aux guillemets d'échappement dans la chaîne JSON varient en fonction de votre terminal. Pour plus d'informations, consultez la section [Utilisation des guillemets dans les chaînes de caractères](#) du Guide de l'utilisateur AWS Command Line Interface .

Vous pouvez également utiliser l'URL d'un fichier en entrée pour l'option `--resources-to-import`, comme indiqué dans l'exemple suivant.

```

--resources-to-import file://ResourcesToImport.txt
```

Si elle aboutit, cette commande renvoie un résultat similaire à ce qui suit.

```
{
  "Id": "arn:aws:cloudformation:us-west-2:12345678910:changeSet/
ImportChangeSet/8ad75b3f-665f-46f6-a200-0b4727a9442e",
  "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
MyParentStack/4e345b70-1281-11ef-b027-027366d8e82b"
}
```

4. Vérifiez le jeu de modifications pour vous assurer que la pile correcte est importée.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Pour lancer l'ensemble de modifications et importer la pile dans la pile parent source, utilisez la commande `execute-change-set` suivante et remplacez le texte de l'espace réservé. Toutes les [balises au niveau de la pile](#) sont appliquées aux ressources importées à ce moment-là. Une fois l'opération d'importation (`IMPORT_COMPLETE`) terminée, la pile est imbriquée avec succès.

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

Note

Il n'est pas nécessaire d'exécuter la détection de l'écart sur la pile parent après cette opération d'importation, car la ressource `AWS::CloudFormation::Stack` est déjà gérée par CloudFormation.

Importez automatiquement AWS des ressources dans une CloudFormation pile

Vous pouvez désormais importer automatiquement des ressources nommées lors de la création ou de la mise à jour de CloudFormation piles. Une ressource nommée est une ressource dotée d'un nom personnalisé. Pour plus d'informations, voir [Type de nom](#) dans la référence du CloudFormation modèle.

Lorsque vous lancez l'importation automatique, CloudFormation vérifie si les ressources existantes correspondent à votre modèle et les importe pendant le déploiement. Pour les piles imbriquées, créez l'ensemble de modifications à partir de la pile racine.

Une fois l'importation terminée et avant d'effectuer les opérations de pile suivantes, nous vous conseillons d'exécuter la détection de l'écart sur les ressources importées. La détection de l'écart garantit que la configuration du modèle correspond à la configuration réelle. Pour de plus amples informations, veuillez consulter [Déterminez la dérive sur l'ensemble d'une CloudFormation pile](#).

Pour importer une ressource, ils doivent vous conformer aux exigences suivantes :

- La ressource doit avoir un nom personnalisé statique défini dans votre modèle. Les noms dynamiques (utilisant !Ref ou d'autres fonctions) ne sont actuellement pas pris en charge.
- La ressource doit avoir une DeletionPolicy de Retain ou RetainExceptOnCreate.
- La ressource ne doit pas déjà appartenir à une autre CloudFormation pile.
- Le type de ressource doit prendre en charge les opérations CloudFormation d'importation. Pour de plus amples informations, veuillez consulter [Prise en charge des types de ressources](#).
- L'ID principal du type de ressource doit figurer dans le modèle. Les propriétés principales IDs dotées de propriétés en lecture seule ne sont pas prises en charge. Pour savoir quel est l'ID principal d'un type, recherchez la propriété primaryIdentifier dans le schéma de ressource. Pour plus d'informations sur la propriété, consultez [primaryIdentifier](#).

Exemple Exemple d'importation automatique

L'exemple suivant utilise un ensemble de modifications, CreateChangeSet pour créer une pile appelée my-stack basée sur un fichier modèle, template.yaml, et importe automatiquement les ressources correspondantes.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name CreateChangeSet \  
  --change-set-type CREATE \  
  --template-body file://template.yaml \  
  --import-existing-resources
```

Résolution des problèmes

Si l'importation automatique échoue, procédez comme suit pour résoudre le problème :

- Vérifiez que le nom de la ressource dans votre modèle correspond exactement au nom de la ressource
- Vérifiez que la ressource n'est pas déjà gérée par une autre pile

- Assurez-vous que le type de ressource prend en charge les opérations d'importation
- Vérifiez que votre modèle inclut toutes les propriétés requises pour le type de ressource

Rétablissement d'une opération d'importation

Pour rétablir une opération d'importation, spécifiez une politique de suppression `Retain` pour la ressource que vous souhaitez supprimer du modèle afin de vous assurer qu'elle est préservée lorsque vous la supprimez de la pile.

Annuler une opération d'importation à l'aide du AWS Management Console

1. Spécifiez a `Retain` [DeletionPolicy](#) pour les ressources que vous souhaitez supprimer de votre pile. Dans l'exemple de modèle suivant, `GameTable` est la cible de cette opération de rétablissement.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

```
    }
  },
  "GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
      "TableName": "Games",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}
```

2. Ouvrez la CloudFormation console pour effectuer une mise à jour de la pile afin d'appliquer la politique de suppression.
 - a. Dans la page Piles, sélectionnez la pile, choisissez Update (Mettre à jour), puis Update stack (standard) (Mettre à jour la pile (standard)).
 - b. Sous Prepare template (Préparer le modèle), choisissez Replace current template (Remplacer le modèle actuel).
 - c. Dans Specify template (Spécifier le modèle), indiquez le modèle source mis à jour avec l'attribut DeletionPolicy défini sur GamesTable, puis choisissez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL du modèle source mis à jour dans la zone de texte.

- Choisissez Upload a template file (Télécharger un fichier de modèle), puis recherchez le fichier de modèle source mis à jour.
- d. Dans la page Specify stack details (Spécifier les détails de la pile) aucune modification n'est requise. Choisissez Suivant.
 - e. Dans la page Configure stack options (Configurer les options de pile) aucune modification n'est requise. Choisissez Suivant.
 - f. Sur la **MyStack** page Révision, passez en revue vos modifications. Si votre modèle contient des ressources IAM, sélectionnez I acknowledge that this template may create IAM resources (Je sais que ce modèle peut créer des ressources IAM) pour confirmer que vous souhaitez utiliser des ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#). Mettez ensuite à jour votre pile source en créant un jeu de modifications ou mettez à jour votre pile source directement.
3. Supprimez la ressource, les paramètres associés et les sorties du modèle de pile. Dans cet exemple, le modèle ressemble maintenant à ce qui suit.

Exemple JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
```

```

        "ReadCapacityUnits":5,
        "WriteCapacityUnits":1
    }
}
}
}
}

```

4. Répétez l'étape 2 pour supprimer la ressource (GameTable) et ses paramètres et sorties associés de la pile.

Annuler une opération d'importation à l'aide du AWS CLI

1. Spécifiez a Retain [DeletionPolicy](#) pour les ressources que vous souhaitez supprimer de votre pile. Dans l'exemple de modèle suivant, GameTable est la cible de cette opération de rétablissement.

Exemple JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "KeySchema":[
          {
            "AttributeName":"key",
            "KeyType":"HASH"
          }
        ],
        "ProvisionedThroughput":{
          "ReadCapacityUnits":5,
          "WriteCapacityUnits":1
        }
      }
    }
  }
}

```

```

    }
  },
  "GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
      "TableName": "Games",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}

```

2. Mettez à jour la pile pour appliquer la politique de suppression à la ressource.

```
aws cloudformation update-stack --stack-name MyStack
```

3. Supprimez la ressource, les paramètres associés et les sorties du modèle de pile. Dans cet exemple, le modèle ressemble maintenant à ce qui suit.

Exemple JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable":{

```

```
    "Type": "AWS::DynamoDB::Table",
    "Properties": {
      "TableName": "Service",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}
```

4. Mettez à jour la pile pour supprimer la ressource (GameTable) et ses paramètres et sorties associés de la pile.

```
aws cloudformation update-stack --stack-name MyStack
```

Refactorisation de piles

Grâce à la refactorisation des piles, vous pouvez réorganiser les ressources de vos CloudFormation piles tout en préservant les propriétés et les données des ressources existantes. Vous pouvez déplacer des ressources entre des piles, diviser les grandes piles en piles plus petites ou combiner plusieurs piles en une seule.

Rubriques

- [Comment fonctionne la refactorisation de pile](#)
- [Considérations sur la refactorisation des piles](#)
- [Conditions préalables](#)

- [Refactor Stacks \(console\)](#)
- [Piles de refactorisation \(AWS CLI\)](#)
- [Limite de ressources](#)

Comment fonctionne la refactorisation de pile

La refactorisation des piles implique les phases suivantes :

1. Évaluez votre infrastructure actuelle : passez en revue vos CloudFormation stacks et ressources existants afin d'identifier les opportunités de refactoring des stack.
2. Planifiez votre refactorisation : définissez comment les ressources doivent être organisées. Tenez compte de vos dépendances, de vos conventions de dénomination et de vos limites opérationnelles. Cela peut affecter la CloudFormation validation ultérieure.
3. Déterminez les piles de destination : décidez dans quelles piles vous allez refactoriser les ressources. Vous pouvez déplacer des ressources entre au moins 2 piles (à l'aide de la console) et un maximum de 5 piles (à l'AWS CLI aide de). Les ressources peuvent être déplacées entre des piles imbriquées.
4. Mettez à jour vos modèles : modifiez vos CloudFormation modèles pour refléter le changement prévu, par exemple en déplaçant les définitions de ressources entre les modèles. Vous pouvez renommer la logique IDs au cours de ce processus.
5. Créez le refactorisateur de pile : fournissez une liste des noms de pile et des modèles que vous souhaitez refactoriser.
6. Vérifiez l'impact de la refactorisation et résolvez les éventuels conflits : CloudFormation valide les modèles que vous fournissez et vérifie les dépendances entre les piles, les types de ressources présentant des problèmes de mise à jour des balises et les conflits d'ID logique des ressources.

Si la validation réussit, CloudFormation un aperçu des actions de refactorisation qui se produiront pendant l'exécution sera généré.

Si la validation échoue, résolvez les problèmes identifiés et réessayez. En cas de conflit, fournissez un mappage d'ID logique de ressource indiquant la source et la destination des ressources en conflit.

7. Exécutez la refactorisation : après avoir vérifié que les modifications correspondent à vos objectifs de refactorisation, terminez la refactorisation de la pile.
8. Surveiller — Suivez l'état d'exécution pour vous assurer que l'opération se termine correctement.

Considérations sur la refactorisation des piles

Lorsque vous refactorisez vos piles, gardez à l'esprit les points suivants :

- La refactorisation de la pile se limite à la réorganisation des ressources existantes. Vous ne pouvez pas créer ou supprimer des ressources, modifier les configurations des ressources, ni modifier ou ajouter de nouveaux paramètres, conditions ou mappages pendant le refactoring. Pour apporter ces modifications, mettez d'abord à jour vos piles, puis effectuez la refactorisation des piles.
- Vous ne pouvez pas refactoriser la même ressource en plusieurs piles.
- Vous ne pouvez pas refactoriser les ressources qui font référence à des pseudo-paramètres dont les valeurs diffèrent entre les piles source et de destination, par exemple `AWS::StackName`.
- CloudFormation ne supporte pas les piles vides. Si la refactorisation laisse une pile sans ressources, vous devez d'abord ajouter au moins une ressource à cette pile avant de l'exécuter. [create-stack-refactor](#) Il peut s'agir d'une ressource simple comme `AWS::SNS::Topic` ou `AWS::CloudFormation::WaitCondition`. Par exemple :

```
Resources:
  MySimpleSNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: MySimpleTopic
```

- La refactorisation des piles ne prend pas en charge les piles auxquelles sont associées des politiques de pile, indépendamment de ce que ces politiques autorisent ou refusent.

Conditions préalables

Pour refactoriser les piles, vous devez déjà avoir créé les modèles révisés.

Utilisez la commande [get-template](#) pour récupérer les CloudFormation modèles des piles que vous souhaitez refactoriser.

```
aws cloudformation get-template --stack-name Stack1
```

Une fois que vous disposez des modèles, utilisez l'environnement de développement intégré (IDE) de votre choix pour les mettre à jour afin d'utiliser la structure et l'organisation des ressources souhaitées.

Refactor Stacks (console)

Pour refactoriser les piles à l'aide de la console, procédez comme suit.

Pour refactoriser des piles

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez l' Région AWS emplacement de vos piles.
3. Dans le volet de navigation de gauche, choisissez Stack refactors.
4. Sur la page Stack refactor, choisissez Start stack refactor.
5. Dans Description, fournissez une description pour vous aider à identifier votre refactorisateur de pile. Ensuite, choisissez Suivant.
6. Pour la pile 1, procédez comme suit :

- a. Choisissez Mettre à jour le modèle pour une pile existante ou Créer une nouvelle pile.

Si vous avez choisi Mettre à jour le modèle pour une pile existante, sélectionnez une pile existante dans la liste. Ou choisissez Enter a stack ARN pour saisir l'ARN d'une pile existante.

Si vous avez choisi Créer une nouvelle pile, saisissez le nom de la nouvelle pile dans Nom de la pile.

- b. Sous Remplacer le modèle existant par un modèle refactorisé, choisissez l'URL Amazon S3 ou Téléversez un fichier modèle pour télécharger le modèle souhaité pour la pile 1.
- c. Choisissez Suivant.

7. Pour la pile 2, procédez comme suit :

- a. Choisissez Mettre à jour le modèle pour une pile existante ou Créer une nouvelle pile.

Si vous avez choisi Mettre à jour le modèle pour une pile existante, sélectionnez une pile existante dans la liste. Ou choisissez Enter a stack ARN pour saisir l'ARN d'une pile existante.

Si vous avez choisi Créer une nouvelle pile, saisissez le nom de la nouvelle pile dans Nom de la pile.

- b. Sous Remplacer le modèle existant par un modèle refactorisé, choisissez l'URL Amazon S3 ou Charger le fichier modèle pour télécharger le modèle souhaité pour la pile 2.
 - c. Choisissez Suivant.
8. Sur la page Spécifier les identifiants de ressources logiques renommés, assurez-vous de savoir CloudFormation comment refactoriser les piles en mappant toutes les ressources affichées à leur logique correcte. IDs Dans le cadre du refactor de pile, si la logique d'une ressource a IDs été modifiée, vous devez spécifier comment elle a été renommée en fournissant le nom de la pile source, l'ID logique d'origine, le nom de la pile de destination et l'ID logique renommé. Dans certains cas, la CloudFormation console peut détecter automatiquement le mappage des ressources, et vous pouvez simplement vérifier que le mappage des ressources prérempli est correct avant de continuer.
9. Choisissez Suivant.
10. Sur la page Révision et exécution, passez en revue toutes les sélections effectuées lors des étapes précédentes et vérifiez que tout est correctement configuré.
11. Lorsque vous êtes prêt à refactoriser les piles, choisissez Execute stack refactor.

Piles de refactorisation (AWS CLI)

Les AWS CLI commandes de refactorisation des piles incluent :

- [create-stack-refactor](#) pour valider et générer un aperçu des modifications prévues.
- [describe-stack-refactor](#) pour récupérer le statut et les détails d'une opération de refactorisation de pile.
- [execute-stack-refactor](#) pour terminer l'opération de refactorisation de pile validée.
- [list-stack-refactors](#) pour répertorier toutes les opérations de refactoring de stack de votre compte avec leur statut actuel et des informations de base.
- [list-stack-refactor-actions](#) pour afficher un aperçu des actions spécifiques qui CloudFormation seront effectuées sur chaque pile et ressource lors de l'exécution du refactoring.

Utilisez la procédure suivante pour refactoriser une pile à l'aide de AWS CLI.

Pour refactoriser des piles

1. Utilisez la [create-stack-refactor](#) commande et fournissez les noms des piles et les modèles mis à jour pour les piles à refactoriser. Incluez l'`--enable-stack-creation` option permettant CloudFormation de créer de nouvelles piles si elles n'existent pas déjà.

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation
```

La commande renvoie un `StackRefactorId` que vous utiliserez ultérieurement.

```
{  
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841"  
}
```

Si des conflits sont détectés lors de la validation du modèle (ce que vous pourrez confirmer à l'étape suivante), utilisez la [create-stack-refactor](#) commande avec l'`--resource-mappings` option.

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation \  
  --resource-mappings file://resource-mapping.json
```

Voici un exemple de fichier `resource-mapping.json`.

```
[  
  {  
    "Source": {  
      "StackName": "Stack1",  
      "LogicalResourceId": "MySNSTopic"  
    },  
    "Destination": {  
      "StackName": "Stack2",  
      "LogicalResourceId": "MyLambdaSNSTopic"  
    }  
  }  
]
```

```
    }
  }
]
```

2. Utilisez la [describe-stack-refactor](#) commande pour vous assurer que Status c'est le cas CREATE_COMPLETE. Cela permet de vérifier que la validation est terminée.

```
aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Exemple de sortie :

```
{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
  "StackIds": [
    "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
    "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
  ],
  "ExecutionStatus": "AVAILABLE",
  "Status": "CREATE_COMPLETE"
}
```

3. Utilisez la [list-stack-refactor-actions](#) commande pour prévisualiser les actions spécifiques qui seront effectuées.

```
aws cloudformation list-stack-refactor-actions \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Exemple de sortie :

```
{
  "StackRefactorActions": [
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestLambdaRole",
      "Description": "No configuration changes detected.",
      "Detection": "AUTO",
      "TagResources": [],
    }
  ]
}
```

```

    "UntagResources": [],
    "ResourceMapping": {
      "Source": {
        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "LogicalResourceId": "MyLambdaRole"
      },
      "Destination": {
        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
        "LogicalResourceId": "MyLambdaRole"
      }
    },
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestFunction",
      "Description": "Resource configuration changes will be validated during
refactor execution.",
      "Detection": "AUTO",
      "TagResources": [
        {
          "Key": "aws:cloudformation:stack-name",
          "Value": "Stack2"
        },
        {
          "Key": "aws:cloudformation:logical-id",
          "Value": "MyFunction"
        },
        {
          "Key": "aws:cloudformation:stack-id",
          "Value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
        }
      ],
      "UntagResources": [
        "aws:cloudformation:stack-name",
        "aws:cloudformation:logical-id",
        "aws:cloudformation:stack-id"
      ],
      "ResourceMapping": {
        "Source": {

```

```

        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "LogicalResourceId": "MyFunction"
    },
    "Destination": {
        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
        "LogicalResourceId": "MyFunction"
    }
}
]
}

```

- Après avoir examiné et confirmé vos modifications, utilisez la [execute-stack-refactor](#) commande pour terminer l'opération de refactorisation de la pile.

```

aws cloudformation execute-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

- Utilisez la [describe-stack-refactor](#) commande pour surveiller l'état d'exécution.

```

aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

Exemple de sortie :

```

{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
  "StackIds": [
    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
  ],
  "ExecutionStatus": "SUCCEEDED",
  "Status": "COMPLETE"
}

```

Limite de ressources

- La refactorisation de pile ne prend en charge que les types de ressources avec un `provisioningType` sur `FULLY_MUTABLE`, que vous pouvez vérifier à l'aide de la commande [describe-type](#).
- CloudFormation validera l'éligibilité des ressources lors de la création du refactoriseur et signalera toutes les ressources non prises en charge dans le résultat de la [describe-stack-refactor](#) commande.
- Les ressources suivantes ne sont pas prises en charge pour la refactorisation des piles :
 - `AWS::ACMPCA::Certificate`
 - `AWS::ACMPCA::CertificateAuthority`
 - `AWS::ACMPCA::CertificateAuthorityActivation`
 - `AWS::ApiGateway::BasePathMapping`
 - `AWS::ApiGateway::Method`
 - `AWS::AppConfig::ConfigurationProfile`
 - `AWS::AppConfig::Deployment`
 - `AWS::AppConfig::Environment`
 - `AWS::AppConfig::Extension`
 - `AWS::AppConfig::ExtensionAssociation`
 - `AWS::AppStream::DirectoryConfig`
 - `AWS::AppStream::StackFleetAssociation`
 - `AWS::AppStream::StackUserAssociation`
 - `AWS::AppStream::User`
 - `AWS::BackupGateway::Hypervisor`
 - `AWS::CertificateManager::Certificate`
 - `AWS::CloudFormation::CustomResource`
 - `AWS::CloudFormation::Macro`
 - `AWS::CloudFormation::WaitCondition`
 - `AWS::CloudFormation::WaitConditionHandle`
 - `AWS::CodeDeploy::DeploymentGroup`
 - `AWS::CodePipeline::CustomActionType`

- AWS::Cognito::UserPoolRiskConfigurationAttachment
- AWS::Cognito::UserPoolUICustomizationAttachment
- AWS::Cognito::UserPoolUserToGroupAttachment
- AWS::Config::ConfigRule
- AWS::Config::ConfigurationRecorder
- AWS::Config::DeliveryChannel
- AWS::DataBrew::Dataset
- AWS::DataBrew::Job
- AWS::DataBrew::Project
- AWS::DataBrew::Recipe
- AWS::DataBrew::Ruleset
- AWS::DataBrew::Schedule
- AWS::DataZone::DataSource
- AWS::DataZone::Environment
- AWS::DataZone::EnvironmentBlueprintConfiguration
- AWS::DataZone::EnvironmentProfile
- AWS::DataZone::Project
- AWS::DataZone::SubscriptionTarget
- AWS::DirectoryService::MicrosoftAD
- AWS::DynamoDB::GlobalTable
- AWS::EC2::CustomerGateway
- AWS::EC2::EIP
- AWS::EC2::LaunchTemplate
- AWS::EC2::NetworkInterfacePermission
- AWS::EC2::PlacementGroup
- AWS::EC2::SpotFleet
- AWS::EC2::VPCDHCPOptionsAssociation
- AWS::EC2::VolumeAttachment
- AWS::EMR::Cluster
- AWS::EMR::InstanceFleetConfig

- AWS::EMR::InstanceGroupConfig
- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::ReplicationGroup
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache::SecurityGroupIngress
- AWS::ElasticBeanstalk::ConfigurationTemplate
- AWS::ElasticLoadBalancing::LoadBalancer
- AWS::ElasticLoadBalancingV2::ListenerCertificate
- AWS::Elasticsearch::Domain
- AWS::FIS::ExperimentTemplate
- AWS::Glue::Schema
- AWS::GuardDuty::IPSet
- AWS::GuardDuty::PublishingDestination
- AWS::GuardDuty::ThreatIntelSet
- AWS::IAM::AccessKey
- AWS::IAM::UserToGroupAddition
- AWS::ImageBuilder::Component
- AWS::IoT::PolicyPrincipalAttachment
- AWS::IoT::ThingPrincipalAttachment
- AWS::IoTFleetWise::Campaign
- AWS::IoTWireless::WirelessDeviceImportTask
- AWS::Lambda::EventInvokeConfig
- AWS::Lex::BotVersion
- AWS::M2::Application
- AWS::MSK::Configuration
- AWS::MSK::ServerlessCluster
- AWS::Maester::DocumentType
- AWS::MediaTailor::Channel
- AWS::NeptuneGraph::PrivateGraphEndpoint
- AWS::Omics::AnnotationStore

- `AWS::Omics::ReferenceStore`
- `AWS::Omics::SequenceStore`
- `AWS::OpenSearchServerless::Collection`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`
- `AWS::Panorama::PackageVersion`
- `AWS::QuickSight::Theme`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::RefactorSpaces::Environment`
- `AWS::RefactorSpaces::Route`
- `AWS::RefactorSpaces::Service`
- `AWS::RoboMaker::RobotApplication`
- `AWS::RoboMaker::SimulationApplication`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SDB::Domain`
- `AWS::SageMaker::InferenceComponent`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`

- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::PortfolioShare`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::StepFunctions::StateMachineVersion`
- `AWS::Synthetics::Canary`
- `AWS::VoiceID::Domain`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`
- `AWS::WorkSpaces::Workspace`

Prise en charge des types de ressources

Le tableau suivant répertorie les types de AWS ressources qui prennent actuellement en charge les opérations d'importation, de détection de dérive et de génération d'infrastructure sous forme de code (IaC). Chaque nom de type de ressource renvoie à la rubrique de référence correspondante dans le [Guide de référence des modèles CloudFormation](#).

Une ressource qui prend en charge l'importation de ressources peut également prendre en charge l'importation automatique. Pour de plus amples informations, veuillez consulter [Importer AWS des ressources dans une CloudFormation pile](#).

Cette liste de AWS ressources n'est pas exhaustive. Si un type de ressource spécifique ne figure pas dans la liste, cela signifie probablement qu'il n'est pas accessible via l' API de commande du Cloud AWS. Pour plus d'informations, consultez [Types de ressources prenant en charge l'API Cloud Control](#) dans le Guide de l'utilisateur de l'API Cloud Control. Chaque AWS service détermine indépendamment les types de ressources à rendre accessibles via l'API Cloud Control.

CloudFormation prend également en charge les opérations d'importation et de détection de dérive pour les types de ressources privées provisionnables (celles dont le type de provisionnement est l'un FULLY_MUTABLE ou IMMUTABLE l'autre). Pour importer ou effectuer une détection des écarts sur un type de ressource privée, vous devez d'abord enregistrer la version par défaut de ce type de ressource dans votre compte et vous assurer qu'elle peut être provisionnée. Pour de plus amples informations, veuillez consulter [Utilisez des extensions privées tierces qui ont été partagées avec vous](#).

Notez que le générateur iAC ne prend en charge que les AWS ressources compatibles avec l'API Cloud Control de votre région.

Pour accéder par programmation aux informations sur les types de ressources publiques et privées provisionnables, vous pouvez utiliser l' API de commande du Cloud AWS. Pour plus d'informations, consultez [Déterminer si un type de ressource prend en charge l'API Cloud Control](#) dans le Guide de l'utilisateur de l'API Cloud Control.

Pour commencer à utiliser l'importation, la détection des écarts ou le générateur laC, voici quelques rubriques utiles à consulter :

- [Importer AWS des ressources dans une CloudFormation pile](#)
- [Déectez les modifications de configuration non gérées apportées aux piles et aux ressources grâce à la détection des écarts](#)
- [Générez des modèles à partir de ressources existantes avec le générateur laC](#)

Ressource	Importer	Détection des écarts	Générateur laC
AWS::ACMPCA::Certificate	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ACMPCA::CertificateAuthority</u>	 Oui	 Oui	 Oui
<u>AWS::ACMPCA::CertificateAuthorityActivation</u>	 Oui	 Non	 Non
<u>AWS::ACMPCA::Permission</u>	 Oui	 Oui	 Non
<u>AWS::AIOps::InvestigationGroup</u>	 Oui	 Oui	 Non
<u>AWS::APS::AnomalyDetector</u>	 Oui	 Oui	 Non
<u>AWS::APS::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::APS::RuleGroupsNamespace</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::APS::Scraper</u>	 Oui	 Oui	 Non
<u>AWS::APS::Workspace</u>	 Oui	 Oui	 Oui
<u>AWS::ARCRegionSwitch::Plan</u>	 Oui	 Oui	 Non
<u>AWS::ARCZonalShift::AutoshiftObserverNotificationStatus</u>	 Oui	 Oui	 Non
<u>AWS::ARCZonalShift::ZonalAutoshiftConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::AccessAnalyzer::Analyzer</u>	 Oui	 Oui	 Oui
<u>AWS::AmazonMQ::Broker</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AmazonMQ::Configuration</u>	 Oui	 Oui	 Non
<u>AWS::Amplify::App</u>	 Oui	 Oui	 Oui
<u>AWS::Amplify::Branch</u>	 Oui	 Oui	 Oui
<u>AWS::Amplify::Domain</u>	 Oui	 Oui	 Oui
<u>AWS::AmplifyUIBuilder::Component</u>	 Oui	 Oui	 Oui
<u>AWS::AmplifyUIBuilder::Form</u>	 Oui	 Oui	 Oui
<u>AWS::AmplifyUIBuilder::Theme</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ApiGateway::Account</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::ApiKey</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::Authorizer</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::BasePathMapping</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::BasePathMappingV2</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::ClientCertificate</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::Deployment</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ApiGateway::DocumentationPart</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::DocumentationVersion</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::DomainName</u>	 Oui	 Non	 Non
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::DomainNameV2</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::GatewayResponse</u>	 Oui	 Non	 Non
<u>AWS::ApiGateway::Method</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ApiGateway::Model</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::RequestValidator</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::Resource</u>	 Oui	 Oui	 Non
<u>AWS::ApiGateway::RestApi</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::Stage</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::UsagePlan</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGateway::UsagePlanKey</u>	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ApiGateway::VpcLink</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::Api</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGatewayV2::ApiMapping</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGatewayV2::Authorizer</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::Deployment</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGatewayV2::DomainName</u>	 Oui	 Oui	 Oui
<u>AWS::ApiGatewayV2::Integration</u>	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::Model</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::Route</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::RouteResponse</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::RoutingRule</u>	 Oui	 Oui	 Non
<u>AWS::ApiGatewayV2::VpcLink</u>	 Oui	 Oui	 Oui
<u>AWS::AppConfig::Application</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AppConfig::ConfigurationProfile</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::Deployment</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::DeploymentStrategy</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::Environment</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::Extension</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::ExtensionAssociation</u>	 Oui	 Oui	 Non
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AppFlow::Connector</u>	 Oui	 Oui	 Oui
<u>AWS::AppFlow::ConnectorProfile</u>	 Oui	 Oui	 Oui
<u>AWS::AppFlow::Flow</u>	 Oui	 Oui	 Oui
<u>AWS::AppIntegrations::Application</u>	 Oui	 Oui	 Non
<u>AWS::AppIntegrations::DataIntegration</u>	 Oui	 Oui	 Oui
<u>AWS::AppIntegrations::EventIntegration</u>	 Oui	 Oui	 Oui
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::AppRunner::Service</u>	 Oui	 Oui	 Non
<u>AWS::AppRunner::VpcConnector</u>	 Oui	 Oui	 Non
<u>AWS::AppRunner::VpcIngressConnection</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::AppBlock</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::AppBlockBuilder</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::Application</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::DirectoryConfig</u>	 Oui	 Oui	 Oui
<u>AWS::AppStream::Entitlement</u>	 Oui	 Oui	 Non
<u>AWS::AppStream::ImageBuilder</u>	 Oui	 Oui	 Oui
<u>AWS::AppSync::Api</u>	 Oui	 Oui	 Non
<u>AWS::AppSync::ChannelNamespace</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AppSync::DataSource</u>	 Oui	 Oui	 Non
<u>AWS::AppSync::DomainName</u>	 Oui	 Oui	 Oui
<u>AWS::AppSync::DomainNameApiAssociation</u>	 Oui	 Oui	 Non
<u>AWS::AppSync::FunctionConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::AppSync::GraphQLApi</u>	 Non	 Oui	 Non
<u>AWS::AppSync::Resolver</u>	 Oui	 Oui	 Non
<u>AWS::AppSync::SourceApiAssociation</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::AppTest::TestCase</u>	 Oui	 Oui	 Non
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 Oui	 Oui	 Oui
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 Oui	 Non	 Non
<u>AWS::ApplicationInsights::Application</u>	 Oui	 Oui	 Non
<u>AWS::ApplicationSignals::Discovery</u>	 Oui	 Oui	 Non
<u>AWS::ApplicationSignals::GroupingConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::ApplicationSignals::ServiceLevelObjective</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Athena::CapacityReservation</u>	 Oui	 Oui	 Oui
<u>AWS::Athena::DataCatalog</u>	 Oui	 Oui	 Oui
<u>AWS::Athena::NamedQuery</u>	 Oui	 Oui	 Oui
<u>AWS::Athena::PreparedStatement</u>	 Oui	 Oui	 Oui
<u>AWS::Athena::WorkGroup</u>	 Oui	 Non	 Oui
<u>AWS::AuditManager::Assessment</u>	 Oui	 Non	 Non
<u>AWS::AutoScaling::AutoScalingGroup</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::AutoScaling::LaunchConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::AutoScaling::LifecycleHook</u>	 Oui	 Oui	 Oui
<u>AWS::AutoScaling::ScalingPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::AutoScaling::ScheduledAction</u>	 Oui	 Oui	 Oui
<u>AWS::AutoScaling::WarmPool</u>	 Oui	 Oui	 Non
<u>AWS::B2BI::Capability</u>	 Oui	 Oui	 Non
<u>AWS::B2BI::Partnership</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::B2BI::Profile</u>	 Oui	 Oui	 Non
<u>AWS::B2BI::Transformer</u>	 Oui	 Oui	 Non
<u>AWS::BCMDDataExports::Export</u>	 Oui	 Oui	 Non
<u>AWS::Backup::BackupPlan</u>	 Oui	 Oui	 Non
<u>AWS::Backup::BackupSelection</u>	 Oui	 Non	 Oui
<u>AWS::Backup::BackupVault</u>	 Oui	 Oui	 Non
<u>AWS::Backup::Framework</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 Oui	 Oui	 Non
<u>AWS::Backup::ReportPlan</u>	 Oui	 Oui	 Oui
<u>AWS::Backup::RestoreTestingPlan</u>	 Oui	 Oui	 Non
<u>AWS::Backup::RestoreTestingSelection</u>	 Oui	 Oui	 Non
<u>AWS::Backup::TieringConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::BackupGateway::Hypervisor</u>	 Oui	 Oui	 Non
<u>AWS::Batch::ComputeEnvironment</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Batch::ConsumableResource</u>	 Oui	 Oui	 Non
<u>AWS::Batch::JobDefinition</u>	 Oui	 Oui	 Non
<u>AWS::Batch::JobQueue</u>	 Oui	 Oui	 Oui
<u>AWS::Batch::SchedulingPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::Batch::ServiceEnvironment</u>	 Oui	 Oui	 Non
<u>AWS::Bedrock::Agent</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::AgentAlias</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::AutomatedReasoningPolicy</u>	 Oui	 Oui	 Non
<u>AWS::Bedrock::AutomatedReasoningPolicyVersion</u>	 Oui	 Oui	 Non
<u>AWS::Bedrock::Blueprint</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::DataAutomationProject</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::DataSource</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::Flow</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Bedrock::FlowAlias</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::FlowVersion</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::Guardrail</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::GuardrailVersion</u>	 Oui	 Oui	 Non
<u>AWS::Bedrock::IntelligentPromptRouter</u>	 Oui	 Oui	 Oui
<u>AWS::Bedrock::KnowledgeBase</u>	 Oui	 Non	 Oui
<u>AWS::Bedrock::Prompt</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Bedrock::PromptVersion</u>	 Oui	 Oui	 Oui
<u>AWS::BedrockAgentCore::BrowserCustom</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::BrowserProfile</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::CodeInterpreterCustom</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::Evaluator</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::Gateway</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::GatewayTarget</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::BedrockAgentCore::Memory</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::OnlineEvaluationConfig</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::Policy</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::PolicyEngine</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::Runtime</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::RuntimeEndpoint</u>	 Oui	 Oui	 Non
<u>AWS::BedrockAgentCore::WorkloadIdentity</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::BedrockMantle::Project</u>	 Oui	 Oui	 Non
<u>AWS::Billing::BillingView</u>	 Oui	 Oui	 Non
<u>AWS::BillingConductor::BillingGroup</u>	 Oui	 Oui	 Oui
<u>AWS::BillingConductor::CustomLineItem</u>	 Oui	 Oui	 Oui
<u>AWS::BillingConductor::PricingPlan</u>	 Oui	 Oui	 Oui
<u>AWS::BillingConductor::PricingRule</u>	 Oui	 Oui	 Oui
<u>AWS::Budgets::BudgetsAction</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::CE::AnomalyMonitor</u>	 Oui	 Oui	 Non
<u>AWS::CE::AnomalySubscription</u>	 Oui	 Oui	 Non
<u>AWS::CE::CostCategory</u>	 Oui	 Oui	 Oui
<u>AWS::CUR::ReportDefinition</u>	 Oui	 Oui	 Non
<u>AWS::Cases::CaseRule</u>	 Oui	 Oui	 Non
<u>AWS::Cases::Domain</u>	 Oui	 Oui	 Non
<u>AWS::Cases::Field</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Cases::Layout</u>	 Oui	 Oui	 Non
<u>AWS::Cases::Template</u>	 Oui	 Oui	 Non
<u>AWS::Cassandra::Keyspace</u>	 Oui	 Oui	 Non
<u>AWS::Cassandra::Table</u>	 Oui	 Oui	 Oui
<u>AWS::Cassandra::Type</u>	 Oui	 Oui	 Non
<u>AWS::CertificateManager::Account</u>	 Oui	 Oui	 Non
<u>AWS::Chatbot::CustomAction</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Chatbot::MicrosoftTeamsChannelConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::Chatbot::SlackChannelConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::AnalysisTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::Collaboration</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::ConfiguredTable</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::IdMappingTable</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 Oui	 Oui	 Non
<u>AWS::CleanRooms::Membership</u>	 Oui	 Oui	 Oui
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 Oui	 Oui	 Non
<u>AWS::CleanRoomsML::ConfiguredModelAlgorithm</u>	 Oui	 Oui	 Non
<u>AWS::CleanRoomsML::ConfiguredModelAlgorithmAssociation</u>	 Oui	 Oui	 Non
<u>AWS::CleanRoomsML::TrainingDataset</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::GuardHook</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CloudFormation::HookDefaultVersion</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFormation::HookTypeConfig</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFormation::HookVersion</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFormation::LambdaHook</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::ModuleVersion</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::PublicTypeVersion</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CloudFormation::Publisher</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::ResourceVersion</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::Stack</u>	 Oui	 Non	 Non
<u>AWS::CloudFormation::StackSet</u>	 Oui	 Oui	 Non
<u>AWS::CloudFormation::TypeActivation</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::AnycastIpList</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CloudFront::CachePolicy</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::ConnectionFunction</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::ConnectionGroup</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::ContinuousDeploymentPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::Distribution</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::DistributionTenant</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CloudFront::Function</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::KeyGroup</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::KeyValueStore</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::MonitoringSubscription</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::OriginAccessControl</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::OriginRequestPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::PublicKey</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CloudFront::RealtimeLogConfig</u>	 Oui	 Oui	 Oui
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::TrustStore</u>	 Oui	 Oui	 Non
<u>AWS::CloudFront::VpcOrigin</u>	 Oui	 Oui	 Non
<u>AWS::CloudTrail::Channel</u>	 Oui	 Oui	 Non
<u>AWS::CloudTrail::Dashboard</u>	 Oui	 Oui	 Non
<u>AWS::CloudTrail::EventDataStore</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::CloudTrail::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::CloudTrail::Trail</u>	 Oui	 Oui	 Oui
<u>AWS::CloudWatch::Alarm</u>	 Oui	 Oui	 Oui
<u>AWS::CloudWatch::AlarmMuteRule</u>	 Oui	 Oui	 Non
<u>AWS::CloudWatch::CompositeAlarm</u>	 Oui	 Oui	 Oui
<u>AWS::CloudWatch::Dashboard</u>	 Oui	 Oui	 Non
<u>AWS::CloudWatch::MetricStream</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CodeArtifact::Domain</u>	 Oui	 Oui	 Oui
<u>AWS::CodeArtifact::PackageGroup</u>	 Oui	 Oui	 Non
<u>AWS::CodeArtifact::Repository</u>	 Oui	 Oui	 Oui
<u>AWS::CodeBuild::Fleet</u>	 Oui	 Oui	 Non
<u>AWS::CodeConnections::Connection</u>	 Oui	 Oui	 Non
<u>AWS::CodeDeploy::Application</u>	 Oui	 Oui	 Oui
<u>AWS::CodeDeploy::DeploymentConfig</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CodeDeploy::DeploymentGroup</u>	 Oui	 Oui	 Non
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 Oui	 Oui	 Oui
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 Oui	 Oui	 Non
<u>AWS::CodePipeline::CustomActionType</u>	 Oui	 Oui	 Non
<u>AWS::CodePipeline::Pipeline</u>	 Oui	 Oui	 Non
<u>AWS::CodePipeline::Webhook</u>	 Oui	 Oui	 Non
<u>AWS::CodeStarConnections::Connection</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::CodeStarConnections::RepositoryLink</u>	 Oui	 Oui	 Non
<u>AWS::CodeStarConnections::SyncConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::CodeStarNotifications::NotificationRule</u>	 Oui	 Oui	 Oui
<u>AWS::Cognito::IdentityPool</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::IdentityPoolPrincipalTag</u>	 Oui	 Oui	 Oui
<u>AWS::Cognito::IdentityPoolRoleAttachment</u>	 Oui	 Non	 Non
<u>AWS::Cognito::LogDeliveryConfiguration</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Cognito::ManagedLoginBranding</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::Terms</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPool</u>	 Oui	 Non	 Non
<u>AWS::Cognito::UserPoolClient</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPoolDomain</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPoolGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Cognito::UserPoolResourceServer</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 Oui	 Oui	 Non
<u>AWS::Cognito::UserPoolUser</u>	 Oui	 Non	 Non
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 Oui	 Oui	 Non
<u>AWS::Comprehend::DocumentClassifier</u>	 Oui	 Oui	 Oui
<u>AWS::Comprehend::Flywheel</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ComputeOptimizer::AutomationRule</u>	 Oui	 Oui	 Non
<u>AWS::Config::AggregationAuthorization</u>	 Oui	 Oui	 Oui
<u>AWS::Config::ConfigRule</u>	 Oui	 Oui	 Oui
<u>AWS::Config::ConfigurationAggregator</u>	 Oui	 Oui	 Oui
<u>AWS::Config::ConformancePack</u>	 Oui	 Oui	 Oui
<u>AWS::Config::OrganizationConformancePack</u>	 Oui	 Oui	 Oui
<u>AWS::Config::StoredQuery</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Connect::AgentStatus</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::ApprovedOrigin</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::ContactFlow</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::ContactFlowModule</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::ContactFlowModuleAlias</u>	 Oui	 Oui	 Non
<u>AWS::Connect::ContactFlowModuleVersion</u>	 Oui	 Oui	 Non
<u>AWS::Connect::ContactFlowVersion</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Connect::DataTable</u>	 Oui	 Oui	 Non
<u>AWS::Connect::DataTableAttribute</u>	 Oui	 Oui	 Non
<u>AWS::Connect::DataTableRecord</u>	 Oui	 Oui	 Non
<u>AWS::Connect::EmailAddress</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::EvaluationForm</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::HoursOfOperation</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::Instance</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Connect::InstanceStorageConfig</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::IntegrationAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::Notification</u>	 Oui	 Oui	 Non
<u>AWS::Connect::PhoneNumber</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::PredefinedAttribute</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::Prompt</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::Queue</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::Connect::QuickConnect	 Oui	 Oui	 Oui
AWS::Connect::RoutingProfile	 Oui	 Oui	 Oui
AWS::Connect::Rule	 Oui	 Oui	 Non
AWS::Connect::SecurityKey	 Oui	 Oui	 Oui
AWS::Connect::SecurityProfile	 Oui	 Oui	 Oui
AWS::Connect::TaskTemplate	 Oui	 Oui	 Oui
AWS::Connect::TrafficDistributionGroup	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Connect::User</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::UserHierarchyGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::UserHierarchyStructure</u>	 Oui	 Oui	 Non
<u>AWS::Connect::View</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::ViewVersion</u>	 Oui	 Oui	 Oui
<u>AWS::Connect::Workspace</u>	 Oui	 Oui	 Non
<u>AWS::ConnectCampaigns::Campaign</u>	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ConnectCampaignsV2::Campaign</u>	 Oui	 Oui	 Oui
<u>AWS::ControlTower::EnabledBaseline</u>	 Oui	 Oui	 Non
<u>AWS::ControlTower::EnabledControl</u>	 Oui	 Oui	 Oui
<u>AWS::ControlTower::LandingZone</u>	 Oui	 Oui	 Non
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 Oui	 Oui	 Oui
<u>AWS::CustomerProfiles::Domain</u>	 Oui	 Oui	 Oui
<u>AWS::CustomerProfiles::EventStream</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::CustomerProfiles::EventTrigger</u>	 Oui	 Oui	 Non
<u>AWS::CustomerProfiles::Integration</u>	 Oui	 Oui	 Oui
<u>AWS::CustomerProfiles::ObjectType</u>	 Oui	 Oui	 Oui
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 Oui	 Oui	 Non
<u>AWS::DMS::DataMigration</u>	 Oui	 Oui	 Non
<u>AWS::DMS::DataProvider</u>	 Oui	 Oui	 Non
<u>AWS::DMS::InstanceProfile</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::DMS::MigrationProject</u>	 Oui	 Oui	 Non
<u>AWS::DMS::ReplicationConfig</u>	 Oui	 Oui	 Oui
<u>AWS::DSQL::Cluster</u>	 Oui	 Oui	 Non
<u>AWS::DataBrew::Dataset</u>	 Oui	 Oui	 Oui
<u>AWS::DataBrew::Job</u>	 Oui	 Oui	 Oui
<u>AWS::DataBrew::Project</u>	 Oui	 Oui	 Oui
<u>AWS::DataBrew::Recipe</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DataBrew::Ruleset</u>	 Oui	 Oui	 Oui
<u>AWS::DataBrew::Schedule</u>	 Oui	 Oui	 Oui
<u>AWS::DataPipeline::Pipeline</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::Agent</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationAzureBlob</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationEFS</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationFSxLustre</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DataSync::LocationFSxONTAP</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationFSxWindows</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationHDFS</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationNFS</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationObjectStorage</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::LocationS3</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DataSync::LocationSMB</u>	 Oui	 Oui	 Oui
<u>AWS::DataSync::Task</u>	 Oui	 Oui	 Oui
<u>AWS::DataZone::Connection</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::DataSource</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::Domain</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::DomainUnit</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::Environment</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::DataZone::EnvironmentActions</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::EnvironmentProfile</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::FormType</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::GroupProfile</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::Owner</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::PolicyGrant</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DataZone::Project</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::ProjectMembership</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::ProjectProfile</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::SubscriptionTarget</u>	 Oui	 Oui	 Non
<u>AWS::DataZone::UserProfile</u>	 Oui	 Oui	 Non
<u>AWS::Deadline::Farm</u>	 Oui	 Oui	 Oui
<u>AWS::Deadline::Fleet</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Deadline::LicenseEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::Deadline::Limit</u>	 Oui	 Oui	 Oui
<u>AWS::Deadline::MeteredProduct</u>	 Oui	 Oui	 Non
<u>AWS::Deadline::Monitor</u>	 Oui	 Oui	 Oui
<u>AWS::Deadline::Queue</u>	 Oui	 Oui	 Oui
<u>AWS::Deadline::QueueEnvironment</u>	 Oui	 Oui	 Non
<u>AWS::Deadline::QueueFleetAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Deadline::QueueLimitAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Deadline::StorageProfile</u>	 Oui	 Oui	 Oui
<u>AWS::Detective::Graph</u>	 Oui	 Oui	 Oui
<u>AWS::Detective::MemberInvitation</u>	 Oui	 Oui	 Oui
<u>AWS::Detective::OrganizationAdmin</u>	 Oui	 Oui	 Oui
<u>AWS::DevOpsAgent::AgentSpace</u>	 Oui	 Oui	 Non
<u>AWS::DevOpsAgent::Association</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DevOpsAgent::Service</u>	 Oui	 Oui	 Non
<u>AWS::DevOpsGuru::LogAnomalyDetectionIntegration</u>	 Oui	 Oui	 Oui
<u>AWS::DevOpsGuru::NotificationChannel</u>	 Oui	 Oui	 Oui
<u>AWS::DevOpsGuru::ResourceCollection</u>	 Oui	 Oui	 Oui
<u>AWS::DeviceFarm::DevicePool</u>	 Oui	 Oui	 Non
<u>AWS::DeviceFarm::InstanceProfile</u>	 Oui	 Oui	 Non
<u>AWS::DeviceFarm::NetworkProfile</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::DeviceFarm::Project</u>	 Oui	 Oui	 Non
<u>AWS::DeviceFarm::TestGridProject</u>	 Oui	 Oui	 Non
<u>AWS::DeviceFarm::VPCEConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::Connection</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::DirectConnectGateway</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::DirectConnectGatewayAssociation</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::Lag</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DirectConnect::PrivateVirtualInterface</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::PublicVirtualInterface</u>	 Oui	 Oui	 Non
<u>AWS::DirectConnect::TransitVirtualInterface</u>	 Oui	 Oui	 Non
<u>AWS::DirectoryService::SimpleAD</u>	 Oui	 Oui	 Oui
<u>AWS::DocDB::GlobalCluster</u>	 Oui	 Oui	 Non
<u>AWS::DocDBElastic::Cluster</u>	 Oui	 Oui	 Oui
<u>AWS::DynamoDB::GlobalTable</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::DynamoDB::Table</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::CapacityManagerDataExport</u>	 Oui	 Oui	 Non
<u>AWS::EC2::CapacityReservation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::CapacityReservationFleet</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::CarrierGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::CustomerGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::DHCPOptions</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::EC2::EC2Fleet</u>	 Oui	 Oui	 Non
<u>AWS::EC2::EIP</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::EIPAssociation</u>	 Oui	 Non	 Oui
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::FlowLog</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::GatewayRouteTableAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::EC2::Host	 Oui	 Oui	 Oui
AWS::EC2::IPAM	 Oui	 Oui	 Oui
AWS::EC2::IPAMAllocation	 Oui	 Oui	 Oui
AWS::EC2::IPAMPool	 Oui	 Oui	 Oui
AWS::EC2::IPAMPoolCidr	 Oui	 Oui	 Oui
AWS::EC2::IPAMPrefixListResolver	 Oui	 Oui	 Non
AWS::EC2::IPAMPrefixListResolverTarget	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::IPAMResourceDiscovery</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::IPAMResourceDiscoveryAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::IPAMScope</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::Instance</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::InstanceConnectEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::InternetGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::IpPoolRouteTableAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::KeyPair</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::LaunchTemplate</u>	 Oui	 Oui	 Non
<u>AWS::EC2::LocalGatewayRoute</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::LocalGatewayRouteTable</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::LocalGatewayVirtualInterface</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::LocalGatewayVirtualInterfaceGroup</u>	 Oui	 Oui	 Non
<u>AWS::EC2::NatGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::NetworkAcl</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 Oui	 Oui	 Non
<u>AWS::EC2::NetworkInsightsAccessScopeAnalysis</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 Oui	 Oui	 Non
<u>AWS::EC2::NetworkInsightsPath</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::NetworkInterface</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::NetworkPerformanceMetricSubscription</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::PlacementGroup</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::PrefixList</u>	 Oui	 Oui	 Non
<u>AWS::EC2::Route</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::RouteServer</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::RouteServerAssociation</u>	 Oui	 Oui	 Non
<u>AWS::EC2::RouteServerEndpoint</u>	 Oui	 Oui	 Non
<u>AWS::EC2::RouteServerPeer</u>	 Oui	 Oui	 Non
<u>AWS::EC2::RouteServerPropagation</u>	 Oui	 Oui	 Non
<u>AWS::EC2::RouteTable</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::SecurityGroup</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::SecurityGroupEgress</u>	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::SecurityGroupIngress</u>	 Oui	 Non	 Non
<u>AWS::EC2::SecurityGroupVpcAssociation</u>	 Oui	 Oui	 Non
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 Oui	 Oui	 Non
<u>AWS::EC2::SpotFleet</u>	 Oui	 Oui	 Non
<u>AWS::EC2::Subnet</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::SubnetCidrBlock</u>	 Oui	 Non	 Oui
<u>AWS::EC2::SubnetNetworkAclAssociation</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 Oui	 Non	 Oui
<u>AWS::EC2::TrafficMirrorFilter</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TrafficMirrorSession</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TrafficMirrorTarget</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGateway</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayAttachment</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::TransitGatewayConnect</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayConnectPeer</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayMeteringPolicy</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayMeteringPolicyEntry</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::TransitGatewayMulticastGroupSource</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayPeeringAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayRoute</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayRouteTable</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::TransitGatewayRouteTableAssociation</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayRouteTablePropagation</u>	 Oui	 Oui	 Non
<u>AWS::EC2::TransitGatewayVpcAttachment</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::EC2::VPC	 Oui	 Oui	 Oui
AWS::EC2::VPCLockPublicAccessExclusion	 Oui	 Oui	 Non
AWS::EC2::VPCLockPublicAccessOptions	 Oui	 Oui	 Non
AWS::EC2::VPCCIDRBlock	 Oui	 Oui	 Non
AWS::EC2::VPCHCPOptionsAssociation	 Oui	 Oui	 Oui
AWS::EC2::VPCEncryptionControl	 Oui	 Oui	 Non
AWS::EC2::VPCEndpoint	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::VPCEndpointConnectionNotification</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::VPCEndpointService</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::VPCEndpointServicePermissions</u>	 Oui	 Oui	 Non
<u>AWS::EC2::VPCGatewayAttachment</u>	 Oui	 Non	 Non
<u>AWS::EC2::VPCPeeringConnection</u>	 Oui	 Oui	 Oui
<u>AWS::EC2::VPNConcentrator</u>	 Oui	 Oui	 Non
<u>AWS::EC2::VPNConnection</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::EC2::VPNConnectionRoute	 Oui	 Oui	 Oui
AWS::EC2::VPNGateway	 Oui	 Oui	 Oui
AWS::EC2::VerifiedAccessEndpoint	 Oui	 Oui	 Oui
AWS::EC2::VerifiedAccessGroup	 Oui	 Oui	 Oui
AWS::EC2::VerifiedAccessInstance	 Oui	 Oui	 Oui
AWS::EC2::VerifiedAccessTrustProvider	 Oui	 Oui	 Oui
AWS::EC2::Volume	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EC2::VolumeAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::ECR::PublicRepository</u>	 Oui	 Oui	 Oui
<u>AWS::ECR::PullThroughCacheRule</u>	 Oui	 Oui	 Oui
<u>AWS::ECR::PullTimeUpdateExclusion</u>	 Oui	 Oui	 Non
<u>AWS::ECR::RegistryPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::ECR::RegistryScanningConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::ECR::ReplicationConfiguration</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ECR::Repository</u>	 Oui	 Oui	 Oui
<u>AWS::ECR::RepositoryCreationTemplate</u>	 Oui	 Oui	 Non
<u>AWS::ECR::SigningConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::ECS::CapacityProvider</u>	 Oui	 Oui	 Non
<u>AWS::ECS::Cluster</u>	 Oui	 Oui	 Oui
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 Oui	 Non	 Oui
<u>AWS::ECS::ExpressGatewayService</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ECS::PrimaryTaskSet</u>	 Oui	 Oui	 Non
<u>AWS::ECS::Service</u>	 Oui	 Oui	 Oui
<u>AWS::ECS::TaskDefinition</u>	 Oui	 Oui	 Oui
<u>AWS::ECS::TaskSet</u>	 Oui	 Oui	 Non
<u>AWS::EFS::AccessPoint</u>	 Oui	 Oui	 Oui
<u>AWS::EFS::FileSystem</u>	 Oui	 Oui	 Oui
<u>AWS::EFS::MountTarget</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EKS::AccessEntry</u>	 Oui	 Oui	 Non
<u>AWS::EKS::Addon</u>	 Oui	 Oui	 Oui
<u>AWS::EKS::Capability</u>	 Oui	 Oui	 Non
<u>AWS::EKS::Cluster</u>	 Oui	 Non	 Oui
<u>AWS::EKS::FargateProfile</u>	 Oui	 Oui	 Oui
<u>AWS::EKS::IdentityProviderConfig</u>	 Oui	 Oui	 Oui
<u>AWS::EKS::Nodegroup</u>	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EKS::PodIdentityAssociation</u>	 Oui	 Oui	 Non
<u>AWS::EMR::SecurityConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::EMR::Step</u>	 Oui	 Oui	 Non
<u>AWS::EMR::Studio</u>	 Oui	 Oui	 Oui
<u>AWS::EMR::StudioSessionMapping</u>	 Oui	 Oui	 Oui
<u>AWS::EMR::WALWorkspace</u>	 Oui	 Oui	 Non
<u>AWS::EMRContainers::Endpoint</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EMRContainers::SecurityConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::EMRContainers::VirtualCluster</u>	 Oui	 Oui	 Non
<u>AWS::EMRServerless::Application</u>	 Oui	 Oui	 Oui
<u>AWS::EVS::Environment</u>	 Oui	 Oui	 Non
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 Oui	 Oui	 Non
<u>AWS::ElastiCache::ParameterGroup</u>	 Oui	 Oui	 Non
<u>AWS::ElastiCache::ReplicationGroup</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ElastiCache::ServerlessCache</u>	 Oui	 Oui	 Non
<u>AWS::ElastiCache::SubnetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::ElastiCache::User</u>	 Oui	 Oui	 Oui
<u>AWS::ElastiCache::UserGroup</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticBeanstalk::Application</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ElasticBeanstalk::Environment</u>	 Oui	 Oui	 Non
<u>AWS::ElasticLoadBalancing::LoadBalancer</u>	 Oui	 Oui	 Non
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 Oui	 Oui	 Non
<u>AWS::ElementalInference::Feed</u>	 Oui	 Oui	 Non
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 Oui	 Oui	 Oui
<u>AWS::EntityResolution::IdNamespace</u>	 Oui	 Oui	 Non
<u>AWS::EntityResolution::MatchingWorkflow</u>	 Oui	 Oui	 Oui
<u>AWS::EntityResolution::PolicyStatement</u>	 Oui	 Oui	 Non
<u>AWS::EntityResolution::SchemaMapping</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::EventSchemas::Discoverer</u>	 Oui	 Oui	 Non
<u>AWS::EventSchemas::Registry</u>	 Oui	 Oui	 Non
<u>AWS::EventSchemas::RegistryPolicy</u>	 Oui	 Oui	 Non
<u>AWS::EventSchemas::Schema</u>	 Oui	 Oui	 Non
<u>AWS::Events::ApiDestination</u>	 Oui	 Oui	 Oui
<u>AWS::Events::Archive</u>	 Oui	 Oui	 Oui
<u>AWS::Events::Connection</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Events::Endpoint</u>	 Oui	 Oui	 Oui
<u>AWS::Events::EventBus</u>	 Oui	 Oui	 Oui
<u>AWS::Events::EventBusPolicy</u>	 Oui	 Non	 Non
<u>AWS::Events::Rule</u>	 Oui	 Oui	 Oui
<u>AWS::Evidently::Experiment</u>	 Oui	 Oui	 Non
<u>AWS::Evidently::Feature</u>	 Oui	 Non	 Non
<u>AWS::Evidently::Launch</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Evidently::Project</u>	 Oui	 Oui	 Non
<u>AWS::Evidently::Segment</u>	 Oui	 Oui	 Non
<u>AWS::FIS::ExperimentTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::FIS::TargetAccountConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::FMS::NotificationChannel</u>	 Oui	 Oui	 Non
<u>AWS::FMS::Policy</u>	 Oui	 Non	 Non
<u>AWS::FMS::ResourceSet</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::FSx::DataRepositoryAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::FSx::S3AccessPointAttachment</u>	 Oui	 Oui	 Non
<u>AWS::FinSpace::Environment</u>	 Oui	 Oui	 Oui
<u>AWS::Forecast::Dataset</u>	 Oui	 Oui	 Non
<u>AWS::Forecast::DatasetGroup</u>	 Oui	 Oui	 Non
<u>AWS::FraudDetector::Detector</u>	 Oui	 Non	 Oui
<u>AWS::FraudDetector::EntityType</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::FraudDetector::EventType</u>	 Oui	 Oui	 Oui
<u>AWS::FraudDetector::Label</u>	 Oui	 Oui	 Oui
<u>AWS::FraudDetector::List</u>	 Oui	 Oui	 Oui
<u>AWS::FraudDetector::Outcome</u>	 Oui	 Oui	 Oui
<u>AWS::FraudDetector::Variable</u>	 Oui	 Oui	 Oui
<u>AWS::GameLift::Alias</u>	 Oui	 Oui	 Oui
<u>AWS::GameLift::Build</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::GameLift::ContainerFleet</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::ContainerGroupDefinition</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::Fleet</u>	 Oui	 Oui	 Oui
<u>AWS::GameLift::GameServerGroup</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::GameSessionQueue</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::Location</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::MatchmakingConfiguration</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::GameLift::MatchmakingRuleSet</u>	 Oui	 Oui	 Non
<u>AWS::GameLift::Script</u>	 Oui	 Oui	 Non
<u>AWS::GameLiftStreams::Application</u>	 Oui	 Oui	 Non
<u>AWS::GameLiftStreams::StreamGroup</u>	 Oui	 Oui	 Non
<u>AWS::GlobalAccelerator::Accelerator</u>	 Oui	 Oui	 Oui
<u>AWS::GlobalAccelerator::CrossAccount Attachment</u>	 Oui	 Oui	 Non
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::GlobalAccelerator::Listener</u>	 Oui	 Oui	 Oui
<u>AWS::Glue::Catalog</u>	 Oui	 Oui	 Non
<u>AWS::Glue::Crawler</u>	 Oui	 Non	 Non
<u>AWS::Glue::Database</u>	 Oui	 Oui	 Non
<u>AWS::Glue::IdentityCenterConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::Glue::Integration</u>	 Oui	 Oui	 Non
<u>AWS::Glue::IntegrationResourceProperty</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Glue::Job</u>	 Oui	 Oui	 Non
<u>AWS::Glue::Registry</u>	 Oui	 Oui	 Oui
<u>AWS::Glue::Schema</u>	 Oui	 Oui	 Oui
<u>AWS::Glue::SchemaVersion</u>	 Oui	 Oui	 Oui
<u>AWS::Glue::SchemaVersionMetadata</u>	 Oui	 Oui	 Oui
<u>AWS::Glue::Trigger</u>	 Oui	 Oui	 Non
<u>AWS::Glue::UsageProfile</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Grafana::Workspace</u>	 Oui	 Oui	 Oui
<u>AWS::GreengrassV2::ComponentVersion</u>	 Oui	 Oui	 Non
<u>AWS::GreengrassV2::Deployment</u>	 Oui	 Oui	 Oui
<u>AWS::GroundStation::Config</u>	 Oui	 Oui	 Oui
<u>AWS::GroundStation::DataflowEndpoint Group</u>	 Oui	 Oui	 Oui
<u>AWS::GroundStation::DataflowEndpoint GroupV2</u>	 Oui	 Oui	 Non
<u>AWS::GroundStation::MissionProfile</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::GuardDuty::Detector</u>	 Oui	 Oui	 Oui
<u>AWS::GuardDuty::Filter</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::IPSet</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::Master</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::Member</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::PublishingDestination</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::GuardDuty::ThreatEntitySet</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::ThreatIntelSet</u>	 Oui	 Oui	 Non
<u>AWS::GuardDuty::TrustedEntitySet</u>	 Oui	 Oui	 Non
<u>AWS::HealthImaging::Datastore</u>	 Oui	 Oui	 Oui
<u>AWS::HealthLake::FHIRDatastore</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::Group</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::GroupPolicy</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IAM::InstanceProfile</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::ManagedPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::OIDCProvider</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::Role</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::RolePolicy</u>	 Oui	 Oui	 Non
<u>AWS::IAM::SAMLProvider</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::ServerCertificate</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IAM::ServiceLinkedRole</u>	 Oui	 Oui	 Non
<u>AWS::IAM::User</u>	 Oui	 Oui	 Oui
<u>AWS::IAM::UserPolicy</u>	 Oui	 Oui	 Non
<u>AWS::IAM::VirtualMFADevice</u>	 Oui	 Oui	 Non
<u>AWS::IVS::Channel</u>	 Oui	 Oui	 Oui
<u>AWS::IVS::EncoderConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::IVS::IngestConfiguration</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IVS::PlaybackKeyPair</u>	 Oui	 Oui	 Oui
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 Oui	 Oui	 Non
<u>AWS::IVS::PublicKey</u>	 Oui	 Oui	 Non
<u>AWS::IVS::RecordingConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::IVS::Stage</u>	 Oui	 Oui	 Non
<u>AWS::IVS::StorageConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::IVS::StreamKey</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IVSChat::LoggingConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::IVSChat::Room</u>	 Oui	 Oui	 Oui
<u>AWS::IdentityStore::Group</u>	 Oui	 Oui	 Oui
<u>AWS::IdentityStore::GroupMembership</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::Component</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::ContainerRecipe</u>	 Oui	 Non	 Non
<u>AWS::ImageBuilder::DistributionConfiguration</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ImageBuilder::Image</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::ImagePipeline</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::ImageRecipe</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 Oui	 Oui	 Non
<u>AWS::ImageBuilder::Workflow</u>	 Oui	 Oui	 Non
<u>AWS::Inspector::AssessmentTarget</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Inspector::AssessmentTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::Inspector::ResourceGroup</u>	 Oui	 Oui	 Non
<u>AWS::InspectorV2::CisScanConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::InspectorV2::CodeSecurityIntegration</u>	 Oui	 Oui	 Non
<u>AWS::InspectorV2::CodeSecurityScanConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::InspectorV2::Filter</u>	 Oui	 Oui	 Non
<u>AWS::InternetMonitor::Monitor</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Invoicing::InvoiceUnit</u>	 Oui	 Oui	 Non
<u>AWS::IoT::AccountAuditConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::Authorizer</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::BillingGroup</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::CACertificate</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::Certificate</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::CertificateProvider</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoT::Command</u>	 Oui	 Oui	 Non
<u>AWS::IoT::CustomMetric</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::Dimension</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::DomainConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::IoT::EncryptionConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::IoT::FleetMetric</u>	 Oui	 Oui	 Non
<u>AWS::IoT::JobTemplate</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoT::Logging</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::MitigationAction</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::Policy</u>	 Oui	 Non	 Oui
<u>AWS::IoT::ProvisioningTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::ResourceSpecificLogging</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::RoleAlias</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::ScheduledAudit</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IoT::SecurityProfile</u>	 Oui	 Non	 Oui
<u>AWS::IoT::SoftwarePackage</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::SoftwarePackageVersion</u>	 Oui	 Oui	 Non
<u>AWS::IoT::Thing</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::ThingGroup</u>	 Oui	 Oui	 Oui
<u>AWS::IoT::ThingType</u>	 Oui	 Oui	 Non
<u>AWS::IoT::TopicRule</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoT::TopicRuleDestination</u>	 Oui	 Oui	 Oui
<u>AWS::IoTAnalytics::Channel</u>	 Oui	 Oui	 Oui
<u>AWS::IoTAnalytics::Dataset</u>	 Oui	 Oui	 Oui
<u>AWS::IoTAnalytics::Datastore</u>	 Oui	 Oui	 Oui
<u>AWS::IoTAnalytics::Pipeline</u>	 Oui	 Oui	 Oui
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 Oui	 Oui	 Oui
<u>AWS::IoTEvents::AlarmModel</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoTEvents::DetectorModel</u>	 Oui	 Oui	 Oui
<u>AWS::IoTEvents::Input</u>	 Oui	 Oui	 Oui
<u>AWS::IoTFleetWise::Campaign</u>	 Oui	 Oui	 Oui
<u>AWS::IoTFleetWise::DecoderManifest</u>	 Oui	 Oui	 Oui
<u>AWS::IoTFleetWise::Fleet</u>	 Oui	 Oui	 Oui
<u>AWS::IoTFleetWise::ModelManifest</u>	 Oui	 Non	 Oui
<u>AWS::IoTFleetWise::SignalCatalog</u>	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoT FleetWise::StateTemplate</u>	 Oui	 Oui	 Non
<u>AWS::IoT FleetWise::Vehicle</u>	 Oui	 Oui	 Oui
<u>AWS::IoT SiteWise::AccessPolicy</u>	 Oui	 Oui	 Non
<u>AWS::IoT SiteWise::Asset</u>	 Oui	 Oui	 Oui
<u>AWS::IoT SiteWise::AssetModel</u>	 Oui	 Oui	 Oui
<u>AWS::IoT SiteWise::ComputationModel</u>	 Oui	 Oui	 Non
<u>AWS::IoT SiteWise::Dashboard</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::IoTSiteWise::Dataset</u>	 Oui	 Oui	 Non
<u>AWS::IoTSiteWise::Gateway</u>	 Oui	 Oui	 Oui
<u>AWS::IoTSiteWise::Portal</u>	 Oui	 Oui	 Non
<u>AWS::IoTSiteWise::Project</u>	 Oui	 Oui	 Non
<u>AWS::IoTTwinMaker::ComponentType</u>	 Oui	 Oui	 Oui
<u>AWS::IoTTwinMaker::Entity</u>	 Oui	 Oui	 Oui
<u>AWS::IoTTwinMaker::Scene</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoT TwinMaker::SyncJob</u>	 Oui	 Oui	 Oui
<u>AWS::IoT TwinMaker::Workspace</u>	 Oui	 Oui	 Oui
<u>AWS::IoT Wireless::Destination</u>	 Oui	 Oui	 Oui
<u>AWS::IoT Wireless::DeviceProfile</u>	 Oui	 Oui	 Oui
<u>AWS::IoT Wireless::FirmwareTask</u>	 Oui	 Oui	 Oui
<u>AWS::IoT Wireless::MulticastGroup</u>	 Oui	 Oui	 Oui
<u>AWS::IoT Wireless::NetworkAnalyzerConfiguration</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::IoTWireless::PartnerAccount</u>	 Oui	 Oui	 Oui
<u>AWS::IoTWireless::ServiceProfile</u>	 Oui	 Oui	 Oui
<u>AWS::IoTWireless::TaskDefinition</u>	 Oui	 Oui	 Oui
<u>AWS::IoTWireless::WirelessDevice</u>	 Oui	 Oui	 Oui
<u>AWS::IoTWireless::WirelessDeviceImportTask</u>	 Oui	 Oui	 Oui
<u>AWS::IoTWireless::WirelessGateway</u>	 Oui	 Oui	 Oui
<u>AWS::KMS::Alias</u>	 Oui	 Non	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::KMS::Key</u>	 Oui	 Oui	 Oui
<u>AWS::KMS::ReplicaKey</u>	 Oui	 Non	 Oui
<u>AWS::KafkaConnect::Connector</u>	 Oui	 Non	 Non
<u>AWS::KafkaConnect::CustomPlugin</u>	 Oui	 Oui	 Non
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::Kendra::DataSource</u>	 Oui	 Oui	 Oui
<u>AWS::Kendra::Faq</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Kendra::Index</u>	 Oui	 Oui	 Oui
<u>AWS::KendraRanking::ExecutionPlan</u>	 Oui	 Oui	 Oui
<u>AWS::Kinesis::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::Kinesis::Stream</u>	 Oui	 Oui	 Oui
<u>AWS::Kinesis::StreamConsumer</u>	 Oui	 Oui	 Non
<u>AWS::KinesisAnalyticsV2::Application</u>	 Oui	 Oui	 Non
<u>AWS::KinesisFirehose::DeliveryStream</u>	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::KinesisVideo::SignalingChannel</u>	 Oui	 Oui	 Non
<u>AWS::KinesisVideo::Stream</u>	 Oui	 Oui	 Non
<u>AWS::LakeFormation::DataCellsFilter</u>	 Oui	 Oui	 Non
<u>AWS::LakeFormation::PrincipalPermissions</u>	 Oui	 Oui	 Non
<u>AWS::LakeFormation::Tag</u>	 Oui	 Oui	 Non
<u>AWS::LakeFormation::TagAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Lambda::Alias</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Lambda::CapacityProvider</u>	 Oui	 Oui	 Non
<u>AWS::Lambda::CodeSigningConfig</u>	 Oui	 Oui	 Oui
<u>AWS::Lambda::EventInvokeConfig</u>	 Oui	 Oui	 Non
<u>AWS::Lambda::EventSourceMapping</u>	 Oui	 Non	 Oui
<u>AWS::Lambda::Function</u>	 Oui	 Oui	 Oui
<u>AWS::Lambda::LayerVersion</u>	 Oui	 Oui	 Non
<u>AWS::Lambda::LayerVersionPermission</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Lambda::Permission</u>	 Oui	 Oui	 Oui
<u>AWS::Lambda::Url</u>	 Oui	 Oui	 Oui
<u>AWS::Lambda::Version</u>	 Oui	 Oui	 Oui
<u>AWS::LaunchWizard::Deployment</u>	 Oui	 Oui	 Non
<u>AWS::Lex::Bot</u>	 Oui	 Oui	 Non
<u>AWS::Lex::BotAlias</u>	 Oui	 Oui	 Non
<u>AWS::Lex::BotVersion</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Lex::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::LicenseManager::Grant</u>	 Oui	 Oui	 Non
<u>AWS::LicenseManager::License</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::Alarm</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::Bucket</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::Certificate</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::Container</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Lightsail::Database</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::DatabaseSnapshot</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::Disk</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::DiskSnapshot</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::Distribution</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::Domain</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::Instance</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Lightsail::InstanceSnapshot</u>	 Oui	 Oui	 Non
<u>AWS::Lightsail::LoadBalancer</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 Oui	 Oui	 Oui
<u>AWS::Lightsail::StaticIp</u>	 Oui	 Oui	 Oui
<u>AWS::Location::APIKey</u>	 Oui	 Oui	 Non
<u>AWS::Location::GeofenceCollection</u>	 Oui	 Oui	 Oui
<u>AWS::Location::Map</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Location::PlaceIndex</u>	 Oui	 Oui	 Oui
<u>AWS::Location::RouteCalculator</u>	 Oui	 Oui	 Oui
<u>AWS::Location::Tracker</u>	 Oui	 Oui	 Oui
<u>AWS::Location::TrackerConsumer</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::AccountPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::Delivery</u>	 Oui	 Oui	 Non
<u>AWS::Logs::DeliveryDestination</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Logs::DeliverySource</u>	 Oui	 Oui	 Non
<u>AWS::Logs::Destination</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::Integration</u>	 Oui	 Oui	 Non
<u>AWS::Logs::LogAnomalyDetector</u>	 Oui	 Oui	 Non
<u>AWS::Logs::LogGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::LogStream</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::MetricFilter</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Logs::QueryDefinition</u>	 Oui	 Oui	 Non
<u>AWS::Logs::ResourcePolicy</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::ScheduledQuery</u>	 Oui	 Oui	 Non
<u>AWS::Logs::SubscriptionFilter</u>	 Oui	 Oui	 Oui
<u>AWS::Logs::Transformer</u>	 Oui	 Oui	 Non
<u>AWS::LookoutEquipment::InferenceScheduler</u>	 Oui	 Oui	 Non
<u>AWS::LookoutVision::Project</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::M2::Application	 Oui	 Oui	 Non
AWS::M2::Deployment	 Oui	 Oui	 Non
AWS::M2::Environment	 Oui	 Oui	 Oui
AWS::MPA::ApprovalTeam	 Oui	 Oui	 Non
AWS::MPA::IdentitySource	 Oui	 Oui	 Non
AWS::MSK::BatchScramSecret	 Oui	 Oui	 Oui
AWS::MSK::Cluster	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MSK::ClusterPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::MSK::Configuration</u>	 Oui	 Oui	 Oui
<u>AWS::MSK::Replicator</u>	 Oui	 Oui	 Oui
<u>AWS::MSK::ServerlessCluster</u>	 Oui	 Oui	 Oui
<u>AWS::MSK::Topic</u>	 Oui	 Oui	 Non
<u>AWS::MSK::VpcConnection</u>	 Oui	 Oui	 Oui
<u>AWS::MWAA::Environment</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MWAA::Serverless::Workflow</u>	 Oui	 Oui	 Non
<u>AWS::Macie::AllowList</u>	 Oui	 Oui	 Non
<u>AWS::Macie::CustomDataIdentifier</u>	 Oui	 Oui	 Non
<u>AWS::Macie::FindingsFilter</u>	 Oui	 Oui	 Non
<u>AWS::Macie::Session</u>	 Oui	 Oui	 Oui
<u>AWS::ManagedBlockchain::Accessor</u>	 Oui	 Oui	 Oui
<u>AWS::MediaConnect::Bridge</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MediaConnect::BridgeOutput</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::BridgeSource</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::Flow</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::FlowEntitlement</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::FlowOutput</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::FlowSource</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::FlowVpcInterface</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MediaConnect::Gateway</u>	 Oui	 Oui	 Oui
<u>AWS::MediaConnect::RouterInput</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::RouterNetworkInterface</u>	 Oui	 Oui	 Non
<u>AWS::MediaConnect::RouterOutput</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::ChannelPlacementGroup</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::CloudWatchAlarmTemplate</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::CloudWatchAlarmTemplateGroup</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::MediaLive::Cluster</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::EventBridgeRuleTemplate</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::EventBridgeRuleTemplateGroup</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::Multiplex</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::Multiplexprogram</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::Network</u>	 Oui	 Oui	 Non
<u>AWS::MediaLive::SdiSource</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MediaLive::SignalMap</u>	 Oui	 Oui	 Non
<u>AWS::MediaPackage::Asset</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackage::Channel</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackage::OriginEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackage::PackagingConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackage::PackagingGroup</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackageV2::Channel</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::MediaPackageV2::ChannelGroup</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 Oui	 Oui	 Non
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 Oui	 Oui	 Non
<u>AWS::MediaTailor::Channel</u>	 Oui	 Oui	 Non
<u>AWS::MediaTailor::ChannelPolicy</u>	 Oui	 Oui	 Non
<u>AWS::MediaTailor::LiveSource</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::MediaTailor::SourceLocation</u>	 Oui	 Oui	 Oui
<u>AWS::MediaTailor::VodSource</u>	 Oui	 Oui	 Oui
<u>AWS::MemoryDB::ACL</u>	 Oui	 Oui	 Oui
<u>AWS::MemoryDB::Cluster</u>	 Oui	 Oui	 Oui
<u>AWS::MemoryDB::MultiRegionCluster</u>	 Oui	 Oui	 Non
<u>AWS::MemoryDB::ParameterGroup</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::MemoryDB::SubnetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::MemoryDB::User</u>	 Oui	 Oui	 Oui
<u>AWS::Neptune::DBCluster</u>	 Oui	 Oui	 Oui
<u>AWS::Neptune::DBClusterParameterGroup</u>	 Oui	 Oui	 Non
<u>AWS::Neptune::DBInstance</u>	 Oui	 Oui	 Non
<u>AWS::Neptune::DBParameterGroup</u>	 Oui	 Oui	 Non
<u>AWS::Neptune::DBSubnetGroup</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Neptune::EventSubscription</u>	 Oui	 Oui	 Non
<u>AWS::NeptuneGraph::Graph</u>	 Oui	 Oui	 Non
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 Oui	 Oui	 Non
<u>AWS::NetworkFirewall::Firewall</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::NetworkFirewall::RuleGroup</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 Oui	 Oui	 Non
<u>AWS::NetworkManager::ConnectAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::ConnectPeer</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::CoreNetwork</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::CoreNetworkPrefixListAssociation</u>	 Oui	 Oui	 Non
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::NetworkManager::Device</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 Oui	 Oui	 Non
<u>AWS::NetworkManager::GlobalNetwork</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::Link</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::LinkAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::Site</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::NetworkManager::VpcAttachment</u>	 Oui	 Oui	 Oui
<u>AWS::Notifications::ChannelAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Notifications::EventRule</u>	 Oui	 Oui	 Non
<u>AWS::Notifications::ManagedNotificationAccountContactAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Notifications::ManagedNotificationAdditionalChannelAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Notifications::NotificationConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::Notifications::NotificationHub</u>	 Oui	 Oui	 Non
<u>AWS::Notifications::OrganizationalUnitAssociation</u>	 Oui	 Oui	 Non
<u>AWS::NotificationsContacts::EmailContact</u>	 Oui	 Oui	 Non
<u>AWS::ODB::CloudAutonomousVmCluster</u>	 Oui	 Oui	 Non
<u>AWS::ODB::CloudExadataInfrastructure</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ODB::CloudVmCluster</u>	 Oui	 Oui	 Non
<u>AWS::ODB::OdbNetwork</u>	 Oui	 Oui	 Non
<u>AWS::ODB::OdbPeeringConnection</u>	 Oui	 Oui	 Non
<u>AWS::OSIS::Pipeline</u>	 Oui	 Oui	 Oui
<u>AWS::Oam::Link</u>	 Oui	 Oui	 Oui
<u>AWS::Oam::Sink</u>	 Oui	 Oui	 Oui
<u>AWS::ObservabilityAdmin::OrganizationCentralizationRule</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ObservabilityAdmin::OrganizationTelemetryRule</u>	 Oui	 Oui	 Non
<u>AWS::ObservabilityAdmin::S3TableIntegration</u>	 Oui	 Oui	 Non
<u>AWS::ObservabilityAdmin::TelemetryEnrichment</u>	 Oui	 Oui	 Non
<u>AWS::ObservabilityAdmin::TelemetryPipelines</u>	 Oui	 Oui	 Non
<u>AWS::ObservabilityAdmin::TelemetryRule</u>	 Oui	 Oui	 Non
<u>AWS::Omics::AnnotationStore</u>	 Oui	 Oui	 Oui
<u>AWS::Omics::ReferenceStore</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Omics::RunGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Omics::SequenceStore</u>	 Oui	 Oui	 Oui
<u>AWS::Omics::VariantStore</u>	 Oui	 Oui	 Oui
<u>AWS::Omics::Workflow</u>	 Oui	 Oui	 Oui
<u>AWS::Omics::WorkflowVersion</u>	 Oui	 Oui	 Non
<u>AWS::OpenSearchServerless::AccessPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::OpenSearchServerless::Collection</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::OpenSearchServerless::CollectionGroup</u>	 Oui	 Oui	 Non
<u>AWS::OpenSearchServerless::Index</u>	 Oui	 Oui	 Non
<u>AWS::OpenSearchServerless::LifecyclePolicy</u>	 Oui	 Oui	 Non
<u>AWS::OpenSearchServerless::SecurityConfig</u>	 Oui	 Oui	 Oui
<u>AWS::OpenSearchServerless::SecurityPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::OpenSearchService::Application</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::OpenSearchService::Domain</u>	 Oui	 Oui	 Non
<u>AWS::Organizations::Account</u>	 Oui	 Oui	 Oui
<u>AWS::Organizations::Organization</u>	 Oui	 Oui	 Oui
<u>AWS::Organizations::OrganizationalUnit</u>	 Oui	 Oui	 Oui
<u>AWS::Organizations::Policy</u>	 Oui	 Oui	 Oui
<u>AWS::Organizations::ResourcePolicy</u>	 Oui	 Oui	 Oui
<u>AWS::PCAConectorAD::Connector</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::PCAConnectorAD::DirectoryRegistration</u>	 Oui	 Oui	 Oui
<u>AWS::PCAConnectorAD::ServicePrincipalName</u>	 Oui	 Oui	 Oui
<u>AWS::PCAConnectorAD::Template</u>	 Oui	 Oui	 Oui
<u>AWS::PCAConnectorAD::TemplateGroupAccessControlEntry</u>	 Oui	 Oui	 Oui
<u>AWS::PCAConnectorSCEP::Challenge</u>	 Oui	 Oui	 Non
<u>AWS::PCAConnectorSCEP::Connector</u>	 Oui	 Oui	 Non
<u>AWS::PCS::Cluster</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::PCS::ComputeNodeGroup</u>	 Oui	 Oui	 Non
<u>AWS::PCS::Queue</u>	 Oui	 Oui	 Non
<u>AWS::Panorama::ApplicationInstance</u>	 Oui	 Oui	 Oui
<u>AWS::Panorama::Package</u>	 Oui	 Oui	 Oui
<u>AWS::Panorama::PackageVersion</u>	 Oui	 Oui	 Non
<u>AWS::PaymentCryptography::Alias</u>	 Oui	 Oui	 Non
<u>AWS::PaymentCryptography::Key</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Personalize::Dataset</u>	 Oui	 Oui	 Oui
<u>AWS::Personalize::DatasetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Personalize::Schema</u>	 Oui	 Oui	 Oui
<u>AWS::Personalize::Solution</u>	 Oui	 Oui	 Non
<u>AWS::Pinpoint::InAppTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::Pipes::Pipe</u>	 Oui	 Oui	 Oui
<u>AWS::Proton::EnvironmentAccountConnection</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Proton::EnvironmentTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::Proton::ServiceTemplate</u>	 Oui	 Oui	 Oui
<u>AWS::QBusiness::Application</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::DataAccessor</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::DataSource</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::Index</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::Permission</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::QBusiness::Plugin</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::Retriever</u>	 Oui	 Oui	 Non
<u>AWS::QBusiness::WebExperience</u>	 Oui	 Oui	 Non
<u>AWS::QLDB::Stream</u>	 Oui	 Oui	 Oui
<u>AWS::QuickSight::ActionConnector</u>	 Oui	 Oui	 Non
<u>AWS::QuickSight::Analysis</u>	 Oui	 Oui	 Non
<u>AWS::QuickSight::CustomPermissions</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::QuickSight::Dashboard</u>	 Oui	 Oui	 Non
<u>AWS::QuickSight::DataSet</u>	 Oui	 Oui	 Oui
<u>AWS::QuickSight::DataSource</u>	 Oui	 Oui	 Oui
<u>AWS::QuickSight::Folder</u>	 Oui	 Oui	 Non
<u>AWS::QuickSight::RefreshSchedule</u>	 Oui	 Oui	 Oui
<u>AWS::QuickSight::Template</u>	 Oui	 Oui	 Non
<u>AWS::QuickSight::Theme</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::QuickSight::Topic</u>	 Oui	 Oui	 Oui
<u>AWS::QuickSight::VPCConnection</u>	 Oui	 Oui	 Oui
<u>AWS::RAM::Permission</u>	 Oui	 Oui	 Non
<u>AWS::RAM::ResourceShare</u>	 Oui	 Oui	 Non
<u>AWS::RDS::CustomDBEngineVersion</u>	 Oui	 Oui	 Non
<u>AWS::RDS::DBCluster</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::DBClusterParameterGroup</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::RDS::DBInstance</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::DBParameterGroup</u>	 Oui	 Oui	 Non
<u>AWS::RDS::DBProxy</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::DBProxyEndpoint</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::DBProxyTargetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::DBShardGroup</u>	 Oui	 Oui	 Non
<u>AWS::RDS::DBSubnetGroup</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::RDS::EventSubscription</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::GlobalCluster</u>	 Oui	 Oui	 Oui
<u>AWS::RDS::Integration</u>	 Oui	 Oui	 Non
<u>AWS::RDS::OptionGroup</u>	 Oui	 Oui	 Oui
<u>AWS::RTBFabric::InboundExternalLink</u>	 Oui	 Oui	 Non
<u>AWS::RTBFabric::Link</u>	 Oui	 Oui	 Non
<u>AWS::RTBFabric::OutboundExternalLink</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::RTBFabric::RequesterGateway</u>	 Oui	 Oui	 Non
<u>AWS::RTBFabric::ResponderGateway</u>	 Oui	 Oui	 Non
<u>AWS::RUM::AppMonitor</u>	 Oui	 Oui	 Oui
<u>AWS::Rbin::Rule</u>	 Oui	 Oui	 Non
<u>AWS::Redshift::Cluster</u>	 Oui	 Oui	 Oui
<u>AWS::Redshift::ClusterParameterGroup</u>	 Oui	 Oui	 Non
<u>AWS::Redshift::ClusterSubnetGroup</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Redshift::EndpointAccess</u>	 Oui	 Oui	 Oui
<u>AWS::Redshift::EndpointAuthorization</u>	 Oui	 Oui	 Oui
<u>AWS::Redshift::EventSubscription</u>	 Oui	 Oui	 Non
<u>AWS::Redshift::Integration</u>	 Oui	 Oui	 Non
<u>AWS::Redshift::ScheduledAction</u>	 Oui	 Oui	 Oui
<u>AWS::RedshiftServerless::Namespace</u>	 Oui	 Oui	 Non
<u>AWS::RedshiftServerless::Snapshot</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::RedshiftServerless::Workgroup</u>	 Oui	 Oui	 Non
<u>AWS::RefactorSpaces::Application</u>	 Oui	 Oui	 Oui
<u>AWS::RefactorSpaces::Environment</u>	 Oui	 Oui	 Oui
<u>AWS::RefactorSpaces::Route</u>	 Oui	 Oui	 Oui
<u>AWS::RefactorSpaces::Service</u>	 Oui	 Oui	 Oui
<u>AWS::Rekognition::Collection</u>	 Oui	 Oui	 Oui
<u>AWS::Rekognition::Project</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Rekognition::StreamProcessor</u>	 Oui	 Oui	 Oui
<u>AWS::ResilienceHub::App</u>	 Oui	 Oui	 Oui
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 Oui	 Oui	 Oui
<u>AWS::ResourceExplorer2::DefaultViewAssociation</u>	 Oui	 Oui	 Non
<u>AWS::ResourceExplorer2::Index</u>	 Oui	 Oui	 Oui
<u>AWS::ResourceExplorer2::View</u>	 Oui	 Oui	 Oui
<u>AWS::ResourceGroups::Group</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ResourceGroups::TagSyncTask</u>	 Oui	 Oui	 Non
<u>AWS::RolesAnywhere::CRL</u>	 Oui	 Oui	 Oui
<u>AWS::RolesAnywhere::Profile</u>	 Oui	 Oui	 Oui
<u>AWS::RolesAnywhere::TrustAnchor</u>	 Oui	 Oui	 Non
<u>AWS::Route53::CidrCollection</u>	 Oui	 Oui	 Oui
<u>AWS::Route53::DNSSEC</u>	 Oui	 Oui	 Oui
<u>AWS::Route53::HealthCheck</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Route53::HostedZone</u>	 Oui	 Oui	 Oui
<u>AWS::Route53::KeySigningKey</u>	 Oui	 Oui	 Oui
<u>AWS::Route53GlobalResolver::AccessSource</u>	 Oui	 Oui	 Non
<u>AWS::Route53GlobalResolver::AccessToKen</u>	 Oui	 Oui	 Non
<u>AWS::Route53GlobalResolver::DnsView</u>	 Oui	 Oui	 Non
<u>AWS::Route53GlobalResolver::FirewallDomainList</u>	 Oui	 Oui	 Non
<u>AWS::Route53GlobalResolver::FirewallRule</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Route53GlobalResolver::GlobalResolver</u>	 Oui	 Oui	 Non
<u>AWS::Route53GlobalResolver::HostedZoneAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Route53Profiles::Profile</u>	 Oui	 Oui	 Non
<u>AWS::Route53Profiles::ProfileAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 Oui	 Oui	 Non
<u>AWS::Route53RecoveryControl::Cluster</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryControl::SafetyRule</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryReadiness::Cell</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryReadiness::ReadinessCheck</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryReadiness::RecoveryGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Route53RecoveryReadiness::ResourceSet</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::FirewallDomainList</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Route53Resolver::FirewallRuleGroup</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::FirewallRuleGroupAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::OutpostResolver</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::ResolverConfig</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::ResolverDNSSECConfig</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::ResolverEndpoint</u>	 Oui	 Oui	 Non
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::Route53Resolver::ResolverRule</u>	 Oui	 Oui	 Non
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::S3::AccessGrant</u>	 Oui	 Oui	 Non
<u>AWS::S3::AccessGrantsInstance</u>	 Oui	 Oui	 Non
<u>AWS::S3::AccessGrantsLocation</u>	 Oui	 Oui	 Non
<u>AWS::S3::AccessPoint</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::S3::Bucket</u>	 Oui	 Oui	 Oui
<u>AWS::S3::BucketPolicy</u>	 Oui	 Non	 Oui
<u>AWS::S3::MultiRegionAccessPoint</u>	 Oui	 Oui	 Oui
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 Oui	 Non	 Non
<u>AWS::S3::StorageLens</u>	 Oui	 Oui	 Oui
<u>AWS::S3::StorageLensGroup</u>	 Oui	 Oui	 Non
<u>AWS::S3Express::AccessPoint</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::S3Express::BucketPolicy</u>	 Oui	 Oui	 Non
<u>AWS::S3Express::DirectoryBucket</u>	 Oui	 Oui	 Non
<u>AWS::S3ObjectLambda::AccessPoint</u>	 Oui	 Oui	 Oui
<u>AWS::S3ObjectLambda::AccessPointPolicy</u>	 Oui	 Oui	 Non
<u>AWS::S3Outposts::AccessPoint</u>	 Oui	 Oui	 Oui
<u>AWS::S3Outposts::Bucket</u>	 Oui	 Oui	 Oui
<u>AWS::S3Outposts::BucketPolicy</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::S3Outposts::Endpoint</u>	 Oui	 Oui	 Oui
<u>AWS::S3Tables::Namespace</u>	 Oui	 Oui	 Non
<u>AWS::S3Tables::Table</u>	 Oui	 Oui	 Non
<u>AWS::S3Tables::TableBucket</u>	 Oui	 Oui	 Non
<u>AWS::S3Tables::TableBucketPolicy</u>	 Oui	 Oui	 Non
<u>AWS::S3Tables::TablePolicy</u>	 Oui	 Oui	 Non
<u>AWS::S3Vectors::Index</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::S3Vectors::VectorBucket</u>	 Oui	 Oui	 Non
<u>AWS::S3Vectors::VectorBucketPolicy</u>	 Oui	 Oui	 Non
<u>AWS::SES::ConfigurationSet</u>	 Oui	 Oui	 Non
<u>AWS::SES::ConfigurationSetEventDestination</u>	 Non	 Oui	 Non
<u>AWS::SES::ContactList</u>	 Oui	 Oui	 Non
<u>AWS::SES::CustomVerificationEmailTemplate</u>	 Oui	 Oui	 Non
<u>AWS::SES::DedicatedIpPool</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::SES::EmailIdentity</u>	 Oui	 Oui	 Oui
<u>AWS::SES::MailManagerAddonInstance</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerAddonSubscription</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerAddressList</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerArchive</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerIngressPoint</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerRelay</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::SES::MailManagerRuleSet</u>	 Oui	 Oui	 Non
<u>AWS::SES::MailManagerTrafficPolicy</u>	 Oui	 Oui	 Non
<u>AWS::SES::MultiRegionEndpoint</u>	 Oui	 Oui	 Non
<u>AWS::SES::Template</u>	 Oui	 Oui	 Non
<u>AWS::SES::Tenant</u>	 Oui	 Oui	 Non
<u>AWS::SES::VdmAttributes</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::ConfigurationSet</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SMSVOICE::OptOutList</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::PhoneNumber</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::Pool</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::ProtectConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::SMSVOICE::SenderId</u>	 Oui	 Oui	 Non
<u>AWS::SNS::Subscription</u>	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
AWS::SNS::Topic	 Oui	 Oui	 Oui
AWS::SNS::TopicInlinePolicy	 Oui	 Oui	 Non
AWS::SQS::Queue	 Oui	 Oui	 Oui
AWS::SQS::QueueInlinePolicy	 Oui	 Oui	 Non
AWS::SSM::Association	 Oui	 Oui	 Oui
AWS::SSM::Document	 Oui	 Non	 Non
AWS::SSM::MaintenanceWindow	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SSM::MaintenanceWindowTarget</u>	 Oui	 Oui	 Non
<u>AWS::SSM::MaintenanceWindowTask</u>	 Oui	 Oui	 Non
<u>AWS::SSM::Parameter</u>	 Oui	 Oui	 Non
<u>AWS::SSM::PatchBaseline</u>	 Oui	 Oui	 Non
<u>AWS::SSM::ResourceDataSync</u>	 Oui	 Oui	 Non
<u>AWS::SSM::ResourcePolicy</u>	 Oui	 Oui	 Oui
<u>AWS::SSMContacts::Contact</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SSMContacts::ContactChannel</u>	 Oui	 Oui	 Oui
<u>AWS::SSMContacts::Plan</u>	 Oui	 Oui	 Non
<u>AWS::SSMContacts::Rotation</u>	 Oui	 Oui	 Non
<u>AWS::SSMGuiConnect::Preferences</u>	 Oui	 Oui	 Non
<u>AWS::SSMIncidents::ReplicationSet</u>	 Oui	 Oui	 Oui
<u>AWS::SSMIncidents::ResponsePlan</u>	 Oui	 Oui	 Oui
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::SSMQuickSetup::LifecycleAutomation</u>	 Oui	 Oui	 Non
<u>AWS::SSO::Application</u>	 Oui	 Oui	 Non
<u>AWS::SSO::ApplicationAssignment</u>	 Oui	 Oui	 Non
<u>AWS::SSO::Assignment</u>	 Oui	 Oui	 Non
<u>AWS::SSO::Instance</u>	 Oui	 Oui	 Non
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::App</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SageMaker::AppImageConfig</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::Cluster</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::Device</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::DeviceFleet</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::Domain</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::Endpoint</u>	 Non	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SageMaker::FeatureGroup</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::Image</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::ImageVersion</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::InferenceComponent</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::InferenceExperiment</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::MlflowTrackingServer</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SageMaker::ModelCard</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::ModelPackage</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::ModelPackageGroup</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::MonitoringSchedule</u>	 Oui	 Non	 Non
<u>AWS::SageMaker::PartnerApp</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SageMaker::Pipeline</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::ProcessingJob</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::Project</u>	 Oui	 Oui	 Oui
<u>AWS::SageMaker::Space</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::StudioLifecycleConfig</u>	 Oui	 Oui	 Non
<u>AWS::SageMaker::UserProfile</u>	 Oui	 Oui	 Non
<u>AWS::Scheduler::Schedule</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Scheduler::ScheduleGroup</u>	 Oui	 Oui	 Non
<u>AWS::SecretsManager::ResourcePolicy</u>	 Oui	 Non	 Non
<u>AWS::SecretsManager::RotationSchedule</u>	 Oui	 Oui	 Non
<u>AWS::SecretsManager::Secret</u>	 Oui	 Oui	 Oui
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 Oui	 Non	 Non
<u>AWS::SecurityHub::AggregatorV2</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::AutomationRule</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SecurityHub::AutomationRuleV2</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::ConnectorV2</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::DelegatedAdmin</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::FindingAggregator</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::Hub</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::HubV2</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SecurityHub::Insight</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::OrganizationConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::PolicyAssociation</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::ProductSubscription</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::SecurityControl</u>	 Oui	 Oui	 Non
<u>AWS::SecurityHub::Standard</u>	 Oui	 Oui	 Non
<u>AWS::SecurityLake::AwsLogSource</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::SecurityLake::DataLake</u>	 Oui	 Oui	 Non
<u>AWS::SecurityLake::Subscriber</u>	 Oui	 Oui	 Non
<u>AWS::SecurityLake::SubscriberNotification</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 Oui	 Non	 Non
<u>AWS::ServiceCatalog::LaunchNotificationConstraint</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::LaunchRoleConstraint</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::LaunchTemplateConstraint</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::ServiceCatalog::Portfolio</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::PortfolioPrincipalAssociation</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::PortfolioProductAssociation</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::PortfolioShare</u>	 Oui	 Non	 Non
<u>AWS::ServiceCatalog::ResourceUpdateConstraint</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::ServiceAction</u>	 Oui	 Oui	 Oui
<u>AWS::ServiceCatalog::ServiceActionAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::ServiceCatalog::StackSetConstraint</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::TagOption</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalog::TagOptionAssociation</u>	 Oui	 Oui	 Non
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 Oui	 Oui	 Oui
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 Oui	 Oui	 Oui
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 Oui	 Non	 Non
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 Oui	 Non	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::ServiceDiscovery::Service</u>	 Oui	 Oui	 Non
<u>AWS::Shield::DRTAccess</u>	 Oui	 Oui	 Non
<u>AWS::Shield::ProactiveEngagement</u>	 Oui	 Oui	 Non
<u>AWS::Shield::Protection</u>	 Oui	 Oui	 Non
<u>AWS::Shield::ProtectionGroup</u>	 Oui	 Oui	 Non
<u>AWS::Signer::ProfilePermission</u>	 Oui	 Oui	 Non
<u>AWS::Signer::SigningProfile</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SimSpaceWeaver::Simulation</u>	 Oui	 Oui	 Oui
<u>AWS::StepFunctions::Activity</u>	 Oui	 Oui	 Non
<u>AWS::StepFunctions::StateMachine</u>	 Oui	 Oui	 Oui
<u>AWS::StepFunctions::StateMachineAlias</u>	 Oui	 Oui	 Oui
<u>AWS::StepFunctions::StateMachineVersion</u>	 Oui	 Oui	 Oui
<u>AWS::SupportApp::AccountAlias</u>	 Oui	 Oui	 Oui
<u>AWS::SupportApp::SlackChannelConfiguration</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::SupportApp::SlackWorkspaceConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::Synthetics::Canary</u>	 Oui	 Oui	 Oui
<u>AWS::Synthetics::Group</u>	 Oui	 Oui	 Oui
<u>AWS::SystemsManagerSAP::Application</u>	 Oui	 Oui	 Non
<u>AWS::Timestream::Database</u>	 Oui	 Oui	 Oui
<u>AWS::Timestream::InfluxDBCluster</u>	 Oui	 Oui	 Non
<u>AWS::Timestream::InfluxDBInstance</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::Timestream::ScheduledQuery</u>	 Oui	 Oui	 Non
<u>AWS::Timestream::Table</u>	 Oui	 Oui	 Oui
<u>AWS::Transfer::Agreement</u>	 Oui	 Oui	 Oui
<u>AWS::Transfer::Certificate</u>	 Oui	 Oui	 Oui
<u>AWS::Transfer::Connector</u>	 Oui	 Oui	 Oui
<u>AWS::Transfer::Profile</u>	 Oui	 Oui	 Oui
<u>AWS::Transfer::Server</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Transfer::User</u>	 Oui	 Oui	 Non
<u>AWS::Transfer::WebApp</u>	 Oui	 Oui	 Non
<u>AWS::Transfer::Workflow</u>	 Oui	 Oui	 Oui
<u>AWS::VerifiedPermissions::IdentitySource</u>	 Oui	 Oui	 Oui
<u>AWS::VerifiedPermissions::Policy</u>	 Oui	 Oui	 Oui
<u>AWS::VerifiedPermissions::PolicyStore</u>	 Oui	 Oui	 Oui
<u>AWS::VerifiedPermissions::PolicyTemplate</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::VoiceID::Domain</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::AccessLogSubscription</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::AuthPolicy</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::DomainVerification</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::Listener</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::ResourceConfiguration</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::ResourceGateway</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::VpcLattice::ResourcePolicy</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::Rule</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::Service</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::ServiceNetwork</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 Oui	 Oui	 Non
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::VpcLattice::TargetGroup</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::IPSet</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::LoggingConfiguration</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::RegexPatternSet</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::RuleGroup</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::WebACL</u>	 Oui	 Oui	 Oui
<u>AWS::WAFv2::WebACLAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Wisdom::AIAgent</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::AIAgentVersion</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::AIGuardrail</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::AIGuardrailVersion</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::AIPrompt</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::AIPromptVersion</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::Assistant</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur IaC
<u>AWS::Wisdom::AssistantAssociation</u>	 Oui	 Oui	 Oui
<u>AWS::Wisdom::KnowledgeBase</u>	 Oui	 Oui	 Oui
<u>AWS::Wisdom::MessageTemplate</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::MessageTemplateVersion</u>	 Oui	 Oui	 Non
<u>AWS::Wisdom::QuickResponse</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpaces::ConnectionAlias</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpaces::Workspace</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::WorkSpaces::WorkspacesPool</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpacesThinClient::Environment</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 Oui	 Oui	 Oui

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::WorkSpacesWeb::Portal</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::SessionLogger</u>	 Oui	 Oui	 Non
<u>AWS::WorkSpacesWeb::TrustStore</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 Oui	 Oui	 Oui
<u>AWS::WorkSpacesWeb::UserSettings</u>	 Oui	 Oui	 Oui
<u>AWS::WorkspacesInstances::Volume</u>	 Oui	 Oui	 Non
<u>AWS::WorkspacesInstances::VolumeAssociation</u>	 Oui	 Oui	 Non

Ressource	Importer	Détection des écarts	Générateur laC
<u>AWS::WorkspacesInstances::WorkspaceInstance</u>	 Oui	 Oui	 Non
<u>AWS::XRay::Group</u>	 Oui	 Oui	 Oui
<u>AWS::XRay::ResourcePolicy</u>	 Oui	 Oui	 Oui
<u>AWS::XRay::SamplingRule</u>	 Oui	 Oui	 Non
<u>AWS::XRay::TransactionSearchConfig</u>	 Oui	 Oui	 Non

Utilisez des liens de création rapide pour créer des piles CloudFormation

Les liens de création rapide constituent une méthode simplifiée pour lancer des CloudFormation piles directement depuis la console URLs . CloudFormation En spécifiant l'URL du modèle, le nom de la pile et les paramètres du modèle dans les paramètres de requête de l'URL, vous pouvez préremplir une seule page Créer une pile, ce qui accélère le processus. Cette approche simplifie le processus de création des piles en réduisant le nombre d'étapes dans l'assistant et la quantité d'informations

à saisir par l'utilisateur. Il optimise également la réutilisation des modèles, car vous pouvez en créer plusieurs URLs qui spécifient des valeurs différentes pour le même modèle.

Format d'URL

Le lien de création rapide suit le format d'URL suivant :

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/  
stacks/create/review  
?templateURL=TemplateURL  
&stackName=StackName  
&param_parameterName=parameterValue
```

CloudFormation prend en charge les paramètres de requête d'URL suivants :

URL de modèle

Obligatoire. Le paramètre `templateURL` indique l'URL du modèle stocké dans un compartiment Amazon S3. Pour éviter tout problème avec une URL S3 pré-signée, encodez correctement l'URL.

Formats d'URL S3 pris en charge :

- `https://s3.region-code.amazonaws.com/bucket-name/template-name`
- `https://bucket-name.s3.region-code.amazonaws.com/template-name`
- `https://s3-region-code.amazonaws.com/bucket-name/template-name` (format hérité)

Nom de la pile

Facultatif. Utilisez le `stackName` paramètre pour spécifier le nom de la CloudFormation pile à créer. Un nom de pile ne peut contenir que des caractères alphanumériques (sensibles à la casse) et des traits d'union. Il doit commencer par un caractère alphabétique et ne doit pas dépasser 128 caractères.

Paramètres du modèle

Facultatif. Pour les paramètres du modèle qui ne sont pas du type `NoEcho`, utilisez le format `param_parameterName` dans la chaîne de requête URL. Le paramètre d'URL doit comprendre le préfixe `param_` et le segment de nom du paramètre doit correspondre exactement au nom du paramètre dans le modèle. Par exemple : `param_DBName`.

CloudFormation ignore les paramètres qui n'existent pas dans le modèle, ainsi que tous les paramètres définis avec leurs NoEcho propriétés définies sur des `true` types (généralement, les noms d'utilisateur et les mots de passe). Les paramètres d'URL remplacent les valeurs par défaut spécifiées dans le modèle. Vous pouvez inclure autant de paramètres que nécessaire.

Important

Plutôt que d'intégrer des informations sensibles directement dans vos CloudFormation modèles, nous vous recommandons d'utiliser des paramètres dynamiques dans le modèle de pile pour référencer les informations sensibles stockées et gérées à l'extérieur CloudFormation, par exemple dans le AWS Systems Manager Parameter Store ou AWS Secrets Manager.

Pour de plus amples informations, veuillez consulter la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Tous les noms des paramètres de requête sont sensibles à la casse. Les utilisateurs peuvent remplacer ces valeurs dans la console avant de créer la pile.

Exemple

L'exemple suivant est basé sur le modèle d'[exemple d'instance unique de WordPress base](#). La chaîne de requête inclut le paramètre obligatoire `templateURL` et les paramètres `stackName`, `DBName`, `InstanceType` et `KeyName`.

Pour plus de clarté, des sauts de ligne ont été ajoutés à L'URL suivante.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review
?templateURL=https://s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/
WordPress_Single_Instance.template
&stackName=MyWPBlog
&param_DBName=mywpblog
&param_InstanceType=t2.medium
```

L'URL suivante comprend les mêmes paramètres que l'exemple précédent, mais les sauts de ligne ont été supprimés. Il s'agit du format d'URL réel.

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?
region=us-east-2#/stacks/create/review?templateURL=https://
```



```
s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/WordPress_Single_Instance.template&stackName=MyWPBlog&param_DBName=mywpblog&param_InstanceType=
```

Création d'une pile à l'aide d'un lien de création rapide

Lorsque vous ouvrez un lien de création rapide, vous êtes dirigé vers la CloudFormation console. La console s'ouvre directement sur la page Création rapide d'une pile, les valeurs fournies étant utilisées automatiquement pour les paramètres.

Création d'une pile à l'aide d'un lien de création rapide (console)

1. Sur la page Création rapide d'une pile, pour Modèle, URL du modèle, vérifiez que l'URL du modèle est correcte.
2. Développez la section Afficher le modèle pour vérifier le modèle.
3. Pour Nom de la pile, vérifiez le nom prérempli de la pile.
4. Consultez la section Paramètres. Vérifiez que les valeurs préremplies des paramètres sont correctes. Renseignez tous les paramètres obligatoires qui ne sont pas spécifiés dans l'URL. Modifiez les valeurs préremplies si nécessaire.
5. Vous pouvez ensuite configurer les paramètres suivants :
 - Balises : organisez les ressources à l'aide de paires clé-valeur.
 - Autorisations : choisissez le rôle de service IAM utilisé pour les opérations sur la pile.
 - Options en cas d'échec de la pile : choisissez de restaurer (par défaut) ou de conserver les ressources.
 - Stratégie de pile : contrôlez les autorisations de mise à jour des ressources.
 - Configuration du retour en arrière : configurez le retour en arrière basé sur CloudWatch une alarme.
 - Options de notification : configurez les notifications Amazon SNS pour les événements de pile.
 - Options de création de pile : définissez la durée maximale de création de la pile et activez la protection contre la suppression pour éviter les suppressions accidentelles.

Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#).

6. Pour Capacités, complétez toutes les confirmations requises. Par exemple, si votre modèle contient des ressources IAM, sélectionnez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous souhaitez utiliser des ressources IAM dans le modèle. Pour de plus

amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).

7. (Facultatif) Vous pouvez créer un jeu de modifications pour prévisualiser la configuration de la pile avant sa création. Choisissez Créer un jeu de modifications et suivez les instructions. Pour de plus amples informations, veuillez consulter [Prévisualisation de la configuration de la pile](#).
8. Lorsque vous êtes prêt, choisissez Créer une pile pour lancer la pile, puis surveillez la progression de la création de la pile dans l'onglet Événements. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Exemples de commandes d'opération de CloudFormation pile pour AWS CLI et PowerShell

Les exemples de ligne de commande suivants montrent comment effectuer des CloudFormation actions individuelles avec le AWS CLI et PowerShell. Ces exemples ne comprennent que les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez [cloudformation](#) dans la Référence des commandes en ligne AWS CLI .

Les exemples de ce guide utilisent la convention du backslash (\) pour indiquer qu'une longue ligne de commande se poursuit à la ligne suivante.

Rubriques

- [Annuler la mise à jour d'une pile](#)
- [Poursuite de la restauration d'une mise à jour](#)
- [Création d'une pile](#)
- [Créer une pile qui inclut des transformations](#)
- [Supprimer une pile](#)
- [Décrire les événements d'une pile](#)
- [Décrire une ressource d'une pile](#)
- [Décrire les ressources d'une pile](#)
- [Décrire les piles](#)
- [Obtenir un modèle](#)
- [Répertorier les ressources d'une pile](#)
- [Répertorier les piles](#)

- [Mettre à jour une pile](#)
- [Valider votre modèle](#)
- [Téléchargez des artefacts locaux dans un compartiment S3 avec le AWS CLI](#)

Annuler la mise à jour d'une pile

Utilisez la commande [cancel-update-stack](#) pour annuler la mise à jour d'une pile. Pour de plus amples informations, veuillez consulter [Annuler la mise à jour d'une pile](#).

CLI

AWS CLI

Pour annuler une mise à jour de pile en cours

La commande `cancel-update-stack` suivante annule une mise à jour de la pile sur la pile `myteststack` :

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Pour plus de détails sur l'API, reportez-vous [CancelUpdateStack](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : annule une mise à jour sur la pile spécifiée.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [CancelUpdateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : annule une mise à jour sur la pile spécifiée.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [CancelUpdateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Poursuite de la restauration d'une mise à jour

Utilisez la commande [continue-update-rollback](#) pour poursuivre la restauration d'une mise à jour. Pour de plus amples informations, veuillez consulter [Poursuite de la restauration d'une mise à jour](#).

CLI

AWS CLI

Pour réessayer la restauration d'une mise à jour

L'exemple `continue-update-rollback` suivant reprend une opération de restauration suite à un échec antérieur de la mise à jour de la pile.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Cette commande ne produit aucune sortie.

- Pour plus de détails sur l'API, reportez-vous [ContinueUpdateRollback](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : poursuit l'annulation de la pile nommée, qui doit avoir l'état « UPDATE_ROLLBACK_FAILED ». Si la restauration continue réussit, la pile entrera dans l'état « UPDATE_ROLLBACK_COMPLETE ».

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [ContinueUpdateRollback](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : poursuit l'annulation de la pile nommée, qui doit avoir l'état « UPDATE_ROLLBACK_FAILED ». Si la restauration continue réussit, la pile entrera dans l'état « UPDATE_ROLLBACK_COMPLETE ».

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [ContinueUpdateRollback](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Création d'une pile

Utilisez la commande [create-stack](#) pour créer une pile. Vous devez fournir le nom de la pile, l'emplacement d'un modèle valide et tous les paramètres d'entrée requis. Les noms des clés de paramètre sont sensibles à la casse. Si vous tapez mal le nom d'une clé de paramètre, CloudFormation cela ne crée pas la pile et indique que le modèle ne contient pas ce paramètre.

Les exemples suivants montrent comment créer une nouvelle pile avec le nom, le modèle et les paramètres d'entrée spécifiés.

CLI

AWS CLI

Pour créer une AWS CloudFormation pile

La commande `create-stacks` suivante crée une pile portant le nom `myteststack` à l'aide du modèle `sampletemplate.json` :

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Sortie :

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
```

```
}
```

Pour plus d'informations, consultez Stacks dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous [CreateStack](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : crée une nouvelle pile avec le nom spécifié. Le modèle est analysé à partir du contenu fourni avec des paramètres de personnalisation (PK1 « » et « PK2 » représentent les noms des paramètres déclarés dans le contenu du modèle, « PV1 » et « PV2 » représentent les valeurs de ces paramètres. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle ne sera pas annulée.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Exemple 2 : crée une nouvelle pile avec le nom spécifié. Le modèle est analysé à partir du contenu fourni avec des paramètres de personnalisation (PK1 « » et « PK2 » représentent les noms des paramètres déclarés dans le contenu du modèle, « PV1 » et « PV2 » représentent les valeurs de ces paramètres. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle sera annulée.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
```

```
-Parameter @( $p1, $p2 ) `
-OnFailure "ROLLBACK"
```

Exemple 3 : crée une nouvelle pile avec le nom spécifié. Le modèle est obtenu à partir de l'URL Amazon S3 avec des paramètres de personnalisation (PK1« » représente le nom d'un paramètre déclaré dans le contenu du modèle, PV1 « » représente la valeur du paramètre. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle sera annulée (comme si vous spécifiez - DisableRollback \$false).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 4 : crée une nouvelle pile avec le nom spécifié. Le modèle est obtenu à partir de l'URL Amazon S3 avec des paramètres de personnalisation (PK1« » représente le nom d'un paramètre déclaré dans le contenu du modèle, PV1 « » représente la valeur du paramètre. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle sera annulée (comme si vous spécifiez - DisableRollback \$false). La notification spécifiée AENs recevra les événements publiés liés à la pile.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Pour plus de détails sur l'API, reportez-vous [CreateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : crée une nouvelle pile avec le nom spécifié. Le modèle est analysé à partir du contenu fourni avec des paramètres de personnalisation (PK1« » et « PK2 » représentent les noms des paramètres déclarés dans le contenu du modèle, « PV1 » et « PV2 » représentent les valeurs de ces paramètres. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle ne sera pas annulée.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Exemple 2 : crée une nouvelle pile avec le nom spécifié. Le modèle est analysé à partir du contenu fourni avec des paramètres de personnalisation (PK1« » et « PK2 » représentent les noms des paramètres déclarés dans le contenu du modèle, « PV1 » et « PV2 » représentent les valeurs de ces paramètres. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue »). Si la création de la pile échoue, elle sera annulée.

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( $p1, $p2 ) `
              -OnFailure "ROLLBACK"
```

Exemple 3 : crée une nouvelle pile avec le nom spécifié. Le modèle est obtenu à partir de l'URL Amazon S3 avec des paramètres de personnalisation (PK1« » représente le nom d'un paramètre déclaré dans le contenu du modèle, PV1 « » représente la valeur du paramètre. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle sera annulée (comme si vous spécifiez - DisableRollback \$false).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 4 : crée une nouvelle pile avec le nom spécifié. Le modèle est obtenu à partir de l'URL Amazon S3 avec des paramètres de personnalisation (PK1« » représente le nom d'un

paramètre déclaré dans le contenu du modèle, PV1 « » représente la valeur du paramètre. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Si la création de la pile échoue, elle sera annulée (comme si vous spécifiez - DisableRollback \$false). La notification spécifiée AENs recevra les événements publiés liés à la pile.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Pour plus de détails sur l'API, reportez-vous [CreateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Note

Vous pouvez utiliser AWS CLI `--template-url` cette option pour spécifier l'emplacement d'un fichier modèle dans Amazon S3 ou AWS Systems Manager.

Pour Amazon S3, l'URL doit commencer par `https://`. URLs Les sites Web statiques S3 ne sont pas pris en charge.

```
--template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
```

Pour AWS Systems Manager, utilisez le format suivant :

```
--template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
```

Créer une pile qui inclut des transformations

Utilisez la commande [deploy](#) pour créer une pile qui inclut des transformations. Lorsque vous créez une pile à partir d'un modèle qui inclut des transformateurs, vous devez utiliser un jeu de modifications. La commande `deploy` combine deux étapes (création d'un jeu de modifications et exécution de celui-ci) en une seule commande.

AWS CLI

La commande `deploy` suivante crée une pile avec le nom, le modèle et les paramètres d'entrée spécifiés.

```
aws cloudformation deploy --stack-name myteststack \  
  --template /path_to_template/my-template.json \  
  --parameter-overrides Key1=Value1 Key2=Value2
```

Supprimer une pile

Utilisez la commande [delete-stack](#) pour supprimer une pile. Pour de plus amples informations, veuillez consulter [Supprimer une pile](#).

CLI

AWS CLI

Pour supprimer une pile

L'exemple `delete-stack` suivant supprime la pile spécifiée.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Cette commande ne produit aucune sortie.

- Pour plus de détails sur l'API, reportez-vous [DeleteStack](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : supprime la pile spécifiée.

```
Remove-CFNStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DeleteStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : supprime la pile spécifiée.

```
Remove-CFNStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DeleteStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Si la suppression échoue et renvoie un état DELETE_FAILED, vous pouvez choisir de supprimer la pile de manière forcée à l'aide de l'option `--deletion-mode`. Voici les valeurs suivantes qui peuvent être utilisées avec `deletion-mode` :

- STANDARD : supprime la pile normalement. Il s'agit du mode de suppression par défaut.
- FORCE_DELETE_STACK : supprime la pile et ignore toutes les ressources qui ne peuvent pas être supprimées.

AWS CLI

La commande `delete-stack` suivante supprime de manière forcée la pile *myteststack* à l'aide de la valeur `FORCE_DELETE_STACK` avec le paramètre `deletion-mode` :

```
aws cloudformation delete-stack --stack-name myteststack \  
--deletion-mode FORCE_DELETE_STACK
```

Cette commande ne produit aucune sortie.

Après avoir utilisé `FORCE_DELETE_STACK`, vous pouvez utiliser la commande `list-stack-resources` pour répertorier les ressources qui ont été ignorées pendant le processus de suppression de la pile. Les ressources conservées afficheront l'état `DELETE_SKIPPED`. Pour de plus amples informations, veuillez consulter [Répertorier les ressources d'une pile](#).

Décrire les événements d'une pile

Utilisez la commande [describe-stack-events](#) pour décrire les événements de la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

CLI

AWS CLI

Pour décrire les événements d'une pile

L'exemple `describe-stack-events` suivant affiche les 2 événements les plus récents pour la pile spécifiée.

```
aws cloudformation describe-stack-events \
  --stack-name my-stack \
  --max-items 2

{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.556Z",
      "ResourceStatus": "UPDATE_COMPLETE"
    },
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.127Z",
      "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
    }
  ],
  "NextToken": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackEvents](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie tous les événements associés à une pile spécifiée.

```
Get-CFNStackEvent -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackEvents](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie tous les événements associés à une pile spécifiée.

```
Get-CFNStackEvent -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackEvents](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Décrire une ressource d'une pile

CLI

AWS CLI

Pour obtenir des informations sur une ressource de pile

L'exemple `describe-stack-resource` suivant affiche des informations pour la ressource nommée `MyFunction` dans la pile spécifiée.

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

Sortie :

```
{
  "StackResourceDetail": {
    "StackName": "MyStack",
    "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "MyFunction",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "Metadata": "{}",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
}
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResource](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie la description d'une ressource identifiée dans le modèle associé à la pile spécifiée par l'identifiant logique DBInstance « My ».

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResource](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie la description d'une ressource identifiée dans le modèle associé à la pile spécifiée par l'identifiant logique DBInstance « My ».

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResource](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Décrire les ressources d'une pile

CLI

AWS CLI

Pour obtenir les informations sur une ressource de pile

L'exemple `describe-stack-resources` suivant affiche des informations pour les ressources dans la pile spécifiée.

```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

Sortie :

```
{
  "StackResources": [
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "bucket",
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
      "ResourceType": "AWS::S3::Bucket",
      "Timestamp": "2019-10-02T04:34:11.345Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
      "Timestamp": "2019-10-02T05:34:27.989Z",
      "ResourceStatus": "UPDATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    }
  ]
}
```

```
    }
  },
  {
    "StackName": "my-stack",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "functionRole",
    "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "Timestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResources](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie les descriptions de AWS ressources pour un maximum de 100 ressources associées à la pile spécifiée. Pour obtenir des informations détaillées sur toutes les ressources associées à une pile, utilisez le `Get-CFNStackResourceSummary`, qui prend également en charge la pagination manuelle des résultats.

```
Get-CFNStackResourceList -StackName "myStack"
```

Exemple 2 : renvoie la description de l'instance Amazon EC2 identifiée dans le modèle associé à la pile spécifiée par l'ID logique « `Ec2Instance` ».

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Exemple 3 : renvoie la description d'un maximum de 100 ressources associées à la pile contenant une instance Amazon EC2 identifiée par l'ID d'instance « `i-123456` ». Pour obtenir des informations détaillées sur toutes les ressources associées à une pile, utilisez le `Get-`

CFNStackResourceSummary, qui prend également en charge la pagination manuelle des résultats.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Exemple 4 : renvoie la description de l'instance Amazon EC2 identifiée par l'ID logique « Ec2Instance » dans le modèle pour une pile. La pile est identifiée à l'aide de l'ID de ressource physique d'une ressource qu'elle contient, dans ce cas également une instance Amazon EC2 avec l'ID d'instance « i-123456 ». Une ressource physique différente peut également être utilisée pour identifier la pile en fonction du contenu du modèle, par exemple un compartiment Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResources](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie les descriptions de AWS ressources pour un maximum de 100 ressources associées à la pile spécifiée. Pour obtenir des informations détaillées sur toutes les ressources associées à une pile, utilisez le Get- CFNStackResourceSummary, qui prend également en charge la pagination manuelle des résultats.

```
Get-CFNStackResourceList -StackName "myStack"
```

Exemple 2 : renvoie la description de l'instance Amazon EC2 identifiée dans le modèle associé à la pile spécifiée par l'ID logique « Ec2Instance ».

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Exemple 3 : renvoie la description d'un maximum de 100 ressources associées à la pile contenant une instance Amazon EC2 identifiée par l'ID d'instance « i-123456 ». Pour obtenir des informations détaillées sur toutes les ressources associées à une pile, utilisez le Get- CFNStackResourceSummary, qui prend également en charge la pagination manuelle des résultats.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Exemple 4 : renvoie la description de l'instance Amazon EC2 identifiée par l'ID logique « Ec2Instance » dans le modèle pour une pile. La pile est identifiée à l'aide de l'ID de ressource physique d'une ressource qu'elle contient, dans ce cas également une instance Amazon EC2 avec l'ID d'instance « i-123456 ». Une ressource physique différente peut également être utilisée pour identifier la pile en fonction du contenu du modèle, par exemple un compartiment Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStackResources](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Décrire les piles

CLI

AWS CLI

Pour décrire les AWS CloudFormation piles

La commande `describe-stacks` suivante affiche des informations récapitulatives pour la pile `myteststack` :

```
aws cloudformation describe-stacks --stack-name myteststack
```

Sortie :

```
{  
  "Stacks": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",  
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample  
template showing how to create a publicly accessible S3 bucket. **WARNING** This  
template creates an S3 bucket. You will be billed for the AWS resources used if  
you create a stack from this template.",  
      "Tags": [],  
      "Outputs": [  
        {
```

```
        "Description": "Name of S3 bucket to hold website content",
        "OutputKey": "BucketName",
        "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
    }
],
"StackStatusReason": null,
"CreationTime": "2013-08-23T01:02:15.422Z",
"Capabilities": [],
"StackName": "myteststack",
"StackStatus": "CREATE_COMPLETE",
"DisableRollback": false
}
]
```

Pour plus d'informations, consultez Stacks dans le guide de l'AWS CloudFormation utilisateur.

- Pour plus de détails sur l'API, reportez-vous [DescribeStacks](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK pour Go V2

 Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import (
    "context"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string
```

```
type CloudFormationActions struct {
    CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
// structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
string) StackOutputs {
    output, err := actor.CfnClient.DescribeStacks(ctx,
&cloudformation.DescribeStacksInput{
        StackName: aws.String(stackName),
    })
    if err != nil || len(output.Stacks) == 0 {
        log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
stackName, err)
    }
    stackOutputs := StackOutputs{}
    for _, out := range output.Stacks[0].Outputs {
        stackOutputs[*out.OutputKey] = *out.OutputValue
    }
    return stackOutputs
}
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStacks](#) à la section Référence des AWS SDK pour Go API.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie une collection d'instances Stack décrivant toutes les piles de l'utilisateur.

```
Get-CFNStack
```

Exemple 2 : renvoie une instance Stack décrivant la pile spécifiée

```
Get-CFNStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStacks](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie une collection d'instances Stack décrivant toutes les piles de l'utilisateur.

```
Get-CFNStack
```

Exemple 2 : renvoie une instance Stack décrivant la pile spécifiée

```
Get-CFNStack -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [DescribeStacks](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Par défaut, la commande `describe-stacks` renvoie les valeurs des paramètres. Pour éviter que des valeurs de paramètres sensibles telles que des mots de passe ne soient renvoyées, incluez une `NoEcho` propriété définie sur `TRUE` dans vos CloudFormation modèles.

Important

L'utilisation de l'attribut `NoEcho` ne masque aucune information stockée dans les lieux suivants :

- La section des Metadata modèles. CloudFormation ne transforme, ne modifie ni n'expédie aucune information que vous incluez dans Metadata cette section. Pour de plus amples informations, veuillez consulter [Métadonnées](#).
- La section de modèle Outputs Pour de plus amples informations, veuillez consulter [Sorties](#)
- L'attribut Metadata d'une définition de ressource. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

Nous vous recommandons vivement de ne pas utiliser ces mécanismes pour inclure des informations sensibles, telles que des mots de passe ou des secrets.

⚠ Important

Plutôt que d'intégrer des informations sensibles directement dans vos CloudFormation modèles, nous vous recommandons d'utiliser des paramètres dynamiques dans le modèle de pile pour référencer les informations sensibles stockées et gérées à l'extérieur CloudFormation, par exemple dans le AWS Systems Manager Parameter Store ou AWS Secrets Manager.

Pour de plus amples informations, veuillez consulter la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

Obtenir un modèle

CLI

AWS CLI

Pour afficher le corps du modèle d'une AWS CloudFormation pile

La commande `get-template` suivante affiche le modèle de la pile `myteststack` :

```
aws cloudformation get-template --stack-name myteststack
```

Sortie :

```
{
  "TemplateBody": {
    "AWSTemplateFormatVersion": "2010-09-09",
    "Outputs": {
      "BucketName": {
        "Description": "Name of S3 bucket to hold website content",
        "Value": {
          "Ref": "S3Bucket"
        }
      }
    },
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if
you create a stack from this template.",
    "Resources": {
```

```
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "PublicRead"
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [GetTemplate](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie le modèle associé à la pile spécifiée.

```
Get-CFNTemplate -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [GetTemplate](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie le modèle associé à la pile spécifiée.

```
Get-CFNTemplate -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [GetTemplate](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Répertorier les ressources d'une pile

CLI

AWS CLI

Pour répertorier les ressources d'une pile

La commande suivante affiche la liste des ressources de la pile spécifiée.

```
aws cloudformation list-stack-resources \  
  --stack-name my-stack
```

Sortie :

```
{  
  "StackResourceSummaries": [  
    {  
      "LogicalResourceId": "bucket",  
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",  
      "ResourceType": "AWS::S3::Bucket",  
      "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",  
      "ResourceStatus": "CREATE_COMPLETE",  
      "DriftInformation": {  
        "StackResourceDriftStatus": "IN_SYNC"  
      }  
    },  
    {  
      "LogicalResourceId": "function",  
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",  
      "ResourceType": "AWS::Lambda::Function",  
      "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",  
      "ResourceStatus": "UPDATE_COMPLETE",  
      "DriftInformation": {  
        "StackResourceDriftStatus": "IN_SYNC"  
      }  
    },  
    {  
      "LogicalResourceId": "functionRole",  
      "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",  
      "ResourceType": "AWS::IAM::Role",  
      "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",  
      "ResourceStatus": "CREATE_COMPLETE",  
      "DriftInformation": {  
        "StackResourceDriftStatus": "IN_SYNC"  
      }  
    }  
  ]  
}
```


- Pour plus de détails sur l'API, reportez-vous [ListStackResources](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie les descriptions de toutes les ressources associées à la pile spécifiée.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [ListStackResources](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie les descriptions de toutes les ressources associées à la pile spécifiée.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Pour plus de détails sur l'API, reportez-vous [ListStackResources](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Répertorier les piles

Utilisez la commande [list-stacks](#) pour répertorier les piles. Pour répertorier uniquement les piles avec les codes d'état spécifiés, incluez l'option `--stack-status-filter`. Vous pouvez spécifier un ou plusieurs codes d'état de pile pour l'option `--stack-status-filter`. Pour de plus amples informations, veuillez consulter [Codes d'état de la pile](#).

CLI

AWS CLI

Pour répertorier les AWS CloudFormation piles

La commande `list-stacks` suivante affiche un résumé de toutes les piles ayant le statut `CREATE_COMPLETE` :

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Sortie :

```
[
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket:
Sample template showing how to create a publicly accessible S3 bucket.
**WARNING** This template creates an S3 bucket. You will be billed for the AWS
resources used if you create a stack from this template.",
    "StackStatusReason": null,
    "CreationTime": "2013-08-26T03:27:10.190Z",
    "StackName": "myteststack",
    "StackStatus": "CREATE_COMPLETE"
  }
]
```

- Pour plus de détails sur l'API, reportez-vous [ListStacks](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : renvoie des informations récapitulatives pour toutes les piles.

```
Get-CFNStackSummary
```

Exemple 2 : renvoie des informations récapitulatives pour toutes les piles en cours de création.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Exemple 3 : renvoie des informations récapitulatives pour toutes les piles en cours de création ou de mise à jour.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",
"UPDATE_IN_PROGRESS")
```

- Pour plus de détails sur l'API, reportez-vous [ListStacks](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : renvoie des informations récapitulatives pour toutes les piles.

```
Get-CFNStackSummary
```

Exemple 2 : renvoie des informations récapitulatives pour toutes les piles en cours de création.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Exemple 3 : renvoie des informations récapitulatives pour toutes les piles en cours de création ou de mise à jour.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
"UPDATE_IN_PROGRESS")
```

- Pour plus de détails sur l'API, reportez-vous [ListStacks](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Mettre à jour une pile

Utilisez la commande [update-stack](#) pour mettre à jour directement une pile. Vous spécifiez la pile, les valeurs de paramètre et les fonctionnalités que vous souhaitez mettre à jour et, si vous voulez utiliser un modèle mis à jour, le nom de ce modèle. Pour de plus amples informations, veuillez consulter [Mettre à jour directement les piles](#).

CLI

AWS CLI

Pour mettre à jour les AWS CloudFormation piles

La commande `update-stack` suivante met à jour le modèle et les paramètres d'entrée de la pile `mystack` :

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template --  
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

La commande `update-stack` suivante met uniquement à jour la valeur de paramètre `SubnetIDs` de la pile `mystack`. Si vous ne spécifiez aucune valeur de paramètre, la valeur par défaut spécifiée dans le modèle est utilisée :

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template  
--parameters ParameterKey=KeyPairName,UsePreviousValue=true  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

La commande `update-stack` suivante ajoute deux rubriques de notification de pile à la pile `mystack` :

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --  
notification-arns "arn:aws:sns:us-east-1:123456789012:mytopic1" "arn:aws:sns:us-east-1:123456789012:mytopic2"
```

Pour plus d'informations, consultez la section [AWS CloudFormation Stack Updates](#) dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous [UpdateStack](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : met à jour la pile « `MyStack` » avec le modèle et les paramètres de personnalisation spécifiés. « `PK1` » représente le nom d'un paramètre déclaré dans le modèle et « `PV1` » représente sa valeur. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « `Key` » et « `Value` » au lieu de `ParameterKey` « » et « `ParameterValue` ».

```
Update-CFNStack -StackName "myStack" `  
-TemplateBody "{Template Content Here}" `  
-Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 2 : met à jour la pile « `MyStack` » avec le modèle et les paramètres de personnalisation spécifiés. « `PK1` » et « `PK2` » représentent les noms des paramètres déclarés dans le modèle, « `PV1` » et « `PV2` » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « `Key` » et « `Value` » au lieu de `ParameterKey` « » et « `ParameterValue` ».

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemple 3 : met à jour la pile « MyStack » avec le modèle et les paramètres de personnalisation spécifiés. « PK1 » représente le nom d'un paramètre déclaré dans le modèle et « PV2 » représente sa valeur. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 4 : met à jour la pile « MyStack » avec le modèle spécifié, obtenu auprès d'Amazon S3, et les paramètres de personnalisation. « PK1 » et « PK2 » représentent les noms des paramètres déclarés dans le modèle, « PV1 » et « PV2 » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
    templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemple 5 : met à jour la pile « MyStack », qui est supposée dans cet exemple contenir des ressources IAM, avec le modèle spécifié, obtenu auprès d'Amazon S3, et les paramètres de personnalisation. « PK1 » et « PK2 » représentent les noms des paramètres déclarés dans le modèle, « PV1 » et « PV2 » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Les piles contenant des ressources IAM nécessitent que vous spécifiez le paramètre -Capabilities « CAPABILITY_IAM », sinon la mise à jour échouera avec une erreur « ». InsufficientCapabilities

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
    templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
```

-Capabilities "CAPABILITY_IAM"

- Pour plus de détails sur l'API, reportez-vous [UpdateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : met à jour la pile « MyStack » avec le modèle et les paramètres de personnalisation spécifiés. « PK1 » représente le nom d'un paramètre déclaré dans le modèle et « PV1 » représente sa valeur. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 2 : met à jour la pile « MyStack » avec le modèle et les paramètres de personnalisation spécifiés. « PK1 » et « PK2 » représentent les noms des paramètres déclarés dans le modèle, « PV1 » et « PV2 » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemple 3 : met à jour la pile « MyStack » avec le modèle et les paramètres de personnalisation spécifiés. « PK1 » représente le nom d'un paramètre déclaré dans le modèle et « PV2 » représente sa valeur. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Exemple 4 : met à jour la pile « MyStack » avec le modèle spécifié, obtenu auprès d'Amazon S3, et les paramètres de personnalisation. « PK1 » et « PK2 » représentent les noms des paramètres déclarés dans le modèle, « PV1 » et « PV2 » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ».

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Exemple 5 : met à jour la pile « MyStack », qui est supposée dans cet exemple contenir des ressources IAM, avec le modèle spécifié, obtenu auprès d'Amazon S3, et les paramètres de personnalisation. « PK1 » et « PK2 » représentent les noms des paramètres déclarés dans le modèle, « PV1 » et « PV2 » représentent les valeurs demandées. Les paramètres de personnalisation peuvent également être spécifiés en utilisant « Key » et « Value » au lieu de ParameterKey « » et « ParameterValue ». Les piles contenant des ressources IAM nécessitent que vous spécifiez le paramètre -Capabilities « CAPABILITY_IAM », sinon la mise à jour échouera avec une erreur « ». InsufficientCapabilities

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
                -Capabilities "CAPABILITY_IAM"
```

- Pour plus de détails sur l'API, reportez-vous [UpdateStack](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Note

Pour supprimer toutes les notifications, spécifiez [] pour l'option --notification-arns.

Valider votre modèle

Utilisez la commande [validate-template](#) pour vérifier que votre fichier de modèle ne contient pas d'erreurs de syntaxe.

Lors de la validation, CloudFormation vérifiez d'abord si le modèle est un JSON valide. Si ce n'est pas le cas, CloudFormation vérifie si le modèle est dans un format YAML valide. Si les deux vérifications échouent, CloudFormation renvoie une erreur de validation du modèle.

CLI

AWS CLI

Pour valider un AWS CloudFormation modèle

La commande `validate-template` suivante valide le modèle `sampletemplate.json` :

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Sortie :

```
{
  "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
showing how to create a publicly accessible S3 bucket. **WARNING** This template
creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
  "Parameters": [],
  "Capabilities": []
}
```

Pour plus d'informations, consultez la section Utilisation des AWS CloudFormation modèles dans le guide de AWS CloudFormation l'utilisateur.

- Pour plus de détails sur l'API, reportez-vous [ValidateTemplate](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell V4

Exemple 1 : valide le contenu du modèle spécifié. La sortie détaille les fonctionnalités, la description et les paramètres du modèle.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Exemple 2 : valide le modèle spécifié accessible via une URL Amazon S3. La sortie détaille les fonctionnalités, la description et les paramètres du modèle.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```


- Pour plus de détails sur l'API, reportez-vous [ValidateTemplate](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V4).

Outils pour PowerShell V5

Exemple 1 : valide le contenu du modèle spécifié. La sortie détaille les fonctionnalités, la description et les paramètres du modèle.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Exemple 2 : valide le modèle spécifié accessible via une URL Amazon S3. La sortie détaille les fonctionnalités, la description et les paramètres du modèle.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- Pour plus de détails sur l'API, reportez-vous [ValidateTemplate](#) à la section Référence des Outils AWS pour PowerShell applets de commande (V5).

Voici un exemple de réponse qui génère une erreur de validation.

```
{  
  "ResponseMetadata": {  
    "RequestId": "4ae33ec0-1988-11e3-818b-e15a6df955cd"  
  },  
  "Errors": [  
    {  
      "Message": "Template format error: JSON not well-formed. (line 11, column  
8)",  
      "Code": "ValidationError",  
      "Type": "Sender"  
    }  
  ],  
  "Capabilities": [],  
  "Parameters": []  
}  
A client error (ValidationError) occurred: Template format error: JSON not well-formed.  
(line 11, column 8)
```

 Note

La commande `validate-template` est conçue pour vérifier uniquement la syntaxe de votre modèle. Elle ne garantit pas que les valeurs de propriété que vous avez spécifiées pour une ressource sont valides pour cette ressource. Elle ne détermine pas non plus le nombre de ressources qui existeront lorsque la pile sera créée.

Pour vérifier la validité opérationnelle, vous devez essayer de créer la pile. Il n'y a pas de bac à sable ni de zone de test pour les CloudFormation piles. Les ressources que vous créez lors des tests vous sont donc facturées.

Téléchargez des artefacts locaux dans un compartiment S3 avec le AWS CLI

Vous pouvez utiliser le AWS CLI pour télécharger des artefacts locaux référencés par un CloudFormation modèle dans un compartiment Amazon S3. Les artefacts locaux sont des fichiers auxquels vous faites référence dans votre modèle. Au lieu de télécharger manuellement les fichiers vers un compartiment S3, puis d'ajouter leurs emplacements à votre modèle, vous pouvez spécifier les artefacts locaux dans votre modèle et utiliser la commande [package](#) pour les charger rapidement.

Un artefact local est un chemin vers un fichier ou un dossier que la commande `package` charge dans Amazon S3. Par exemple, un artefact peut être un chemin local vers le code source de votre AWS Lambda fonction ou le OpenAPI fichier d'une API REST Amazon API Gateway.

Lorsque vous utilisez la commande `package` :

- Si vous spécifiez un fichier, la commande le charge directement dans le compartiment S3.
- Si vous spécifiez un dossier, la commande crée un fichier `.zip` pour le dossier, puis charge le fichier `.zip`.
- Si vous ne spécifiez pas de chemin d'accès, la commande crée un fichier `.zip` pour le répertoire de travail et le charge.

Vous pouvez spécifier un chemin d'accès absolu ou relatif, où le chemin d'accès relatif correspond à l'emplacement de votre modèle.

Après avoir chargé les artefacts, la commande renvoie une copie de votre modèle, en remplaçant les références vers des artefacts locaux par l'emplacement S3 où la commande a chargé les artefacts. Vous pouvez ensuite utiliser le modèle renvoyé pour créer ou mettre à jour une pile.

Note

Vous pouvez utiliser des artefacts locaux uniquement pour les propriétés de ressource que la commande `package` prend en charge. Pour plus d'informations sur cette commande et pour obtenir la liste des propriétés de ressources prises en charge, consultez la documentation [package](#) dans la [Référence des commandes AWS CLI](#).

Conditions préalables

Avant de commencer, vous devez disposer d'un compartiment Amazon S3 existant.

package et déployer un modèle avec des artefacts locaux

Le modèle suivant spécifie l'artefact local pour le code source de la fonction Lambda. Le code source est stocké dans le dossier `/home/user/code/lambdafunction`.

Modèle d'origine

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "/home/user/code/lambdafunction"
      }
    }
  }
}
```

La commande [package](#) suivante crée et charge un fichier `.zip` du dossier du code source de la fonction à la racine du compartiment spécifié.

```
aws cloudformation package \  
  --s3-bucket amzn-s3-demo-bucket \  
  --template /path_to_template/template.json \  
  --output-template-file packaged-template.json \  
  --output json
```

La commande génère un nouveau modèle à l'emplacement spécifié par `--output-template-file`. Elle remplace la référence à l'artefact par l'emplacement Amazon S3, comme indiqué ci-dessous. Le `.zip` fichier est nommé en utilisant la MD5 somme de contrôle du contenu du dossier, plutôt que le nom du dossier lui-même.

Modèle obtenu

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Transform": "AWS::Serverless-2016-10-31",  
  "Resources": {  
    "MyFunction": {  
      "Type": "AWS::Serverless::Function",  
      "Properties": {  
        "Handler": "index.handler",  
        "Runtime": "nodejs18.x",  
        "CodeUri": "s3://amzn-s3-demo-bucket/md5 checksum"  
      }  
    }  
  }  
}
```

Après avoir package les artefacts de votre modèle, déployez le modèle traité à l'aide de la commande [deploy](#).

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name
```

Lorsque vous déployez des modèles de plus de 51 200 octets, utilisez la commande `deploy` avec l'option `--s3-bucket` pour charger votre modèle vers S3, comme dans l'exemple suivant.

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --s3-bucket amzn-s3-demo-bucket
```

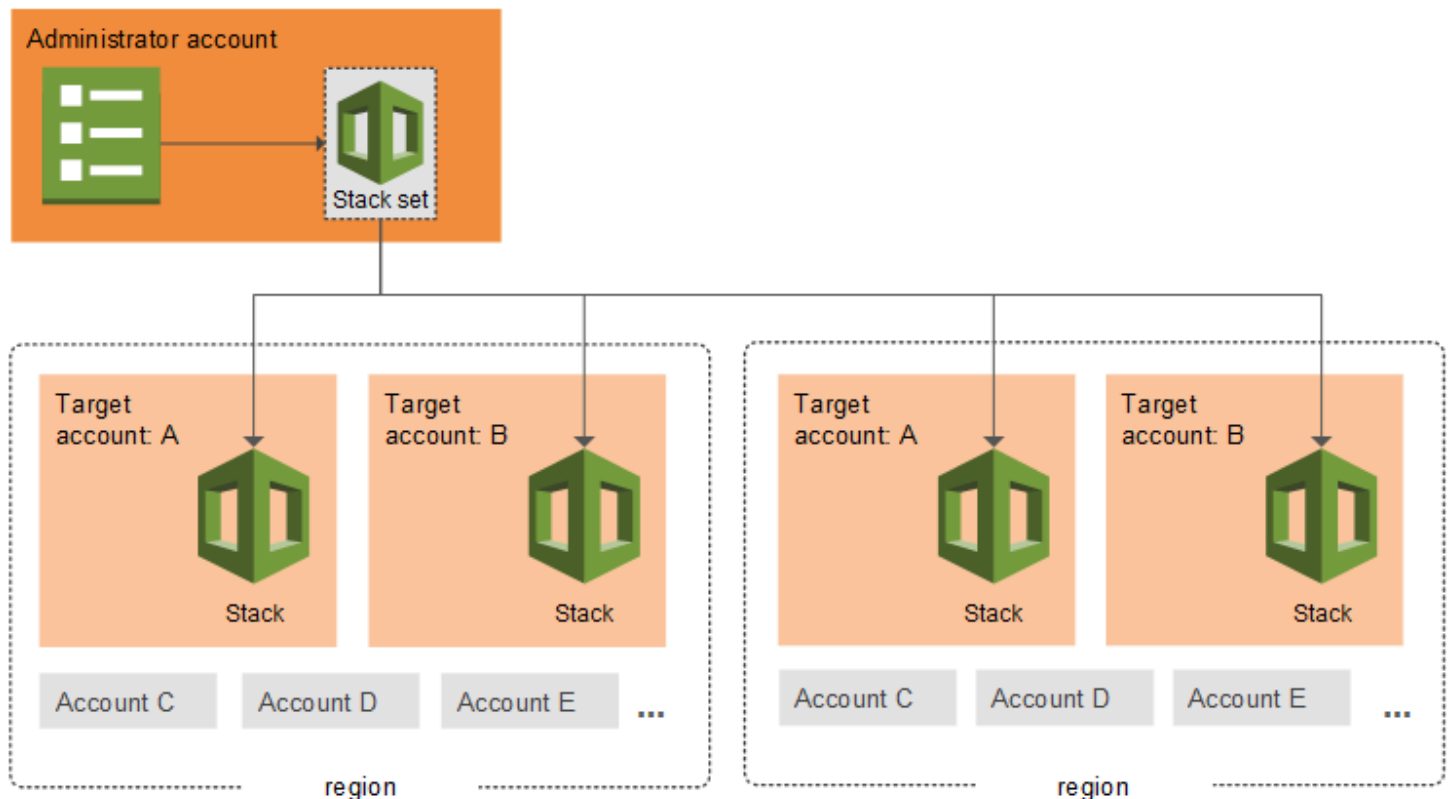
```
--stack-name stack-name \  
--s3-bucket amzn-s3-demo-bucket
```

 Note

Pour un autre exemple d'utilisation de la commande `package` pour charger des artefacts locaux, consultez [Diviser un modèle en éléments réutilisables à l'aide de piles imbriquées](#).

Gérez les piles entre les comptes et les régions avec StackSets

CloudFormation StackSets étend les fonctionnalités des piles en vous permettant de créer, de mettre à jour ou de supprimer des piles sur plusieurs comptes et en Régions AWS une seule opération. À l'aide d'un compte administrateur, vous définissez et gérez un CloudFormation modèle, et vous l'utilisez comme base pour provisionner des piles dans des comptes cibles sélectionnés sur des comptes spécifiques. Régions AWS



Cette section vous aide à commencer à utiliser StackSets et répond aux questions courantes sur la manière d'utiliser et de résoudre les problèmes liés à StackSet la création, aux mises à jour et à la suppression.

Rubriques

- [StackSets concepts](#)
- [Conditions préalables à l'utilisation CloudFormation StackSets](#)
- [Commencez à StackSets utiliser un exemple de modèle](#)
- [Créez CloudFormation StackSets avec des autorisations autogérées](#)

- [Créez CloudFormation StackSets avec des autorisations gérées par le service](#)
- [Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations](#)
- [Mettre à jour CloudFormation StackSets](#)
- [Ajoutez des piles à CloudFormation StackSets](#)
- [Remplacez les valeurs des paramètres sur les piles de votre CloudFormation StackSet](#)
- [Supprimer des piles de CloudFormation StackSets](#)
- [Supprimer CloudFormation StackSets](#)
- [Empêcher les StackSets déploiements échoués à l'aide des portes de compte cible](#)
- [Choisissez le mode de simultanéité pour CloudFormation StackSets](#)
- [Exécution d'une détection de dérive sur CloudFormation StackSets](#)
- [Importez des piles dans CloudFormation StackSets](#)
- [Bonnes pratiques d'utilisation CloudFormation StackSets](#)
- [CloudFormation StackSets exemples de modèles](#)
- [Résolution des problèmes CloudFormation StackSets](#)

StackSets concepts

La terminologie et les concepts suivants sont essentiels à votre compréhension et à votre utilisation de StackSets.

Rubriques

- [Comptes d'administrateur et comptes de destination](#)
- [CloudFormation StackSets](#)
- [Modèles d'autorisation pour StackSets](#)
- [Instances de piles](#)
- [StackSet opérations](#)
- [StackSet options de fonctionnement](#)
- [Étiquettes](#)
- [StackSets codes de statut](#)
- [Codes d'état de l'instance de pile](#)

Comptes d'administrateur et comptes de destination

Le compte administrateur est celui Compte AWS dans lequel vous créez StackSets. Pour StackSets les autorisations gérées par le service, le compte administrateur est soit le compte de gestion de l'organisation, soit un compte d'administrateur délégué. Vous pouvez gérer un StackSet en vous connectant au compte AWS administrateur qui a créé le StackSet.

Un compte cible est le compte dans lequel vous créez, mettez à jour ou supprimez une ou plusieurs piles de votre StackSet compte. Avant de pouvoir utiliser un StackSet pour créer des piles dans un compte cible, établissez une relation de confiance entre l'administrateur et les comptes cibles.

CloudFormation StackSets

A StackSet sert de conteneur pour plusieurs piles déployées dans Comptes AWS des régions spécifiées. Chaque pile est basée sur le même CloudFormation modèle, mais vous pouvez personnaliser des piles individuelles à l'aide de paramètres.

Après avoir défini un StackSet, vous pouvez créer, mettre à jour ou supprimer des piles dans les comptes cibles et Régions AWS vous pouvez le spécifier. Lorsque vous créez, mettez à jour ou supprimez des piles, vous pouvez également spécifier des préférences d'opération. Par exemple, incluez l'ordre des régions dans lesquelles vous souhaitez effectuer l'opération, le seuil de tolérance aux échecs avant l'arrêt des opérations de pile et le nombre de comptes effectuant des opérations de pile simultanément.

A StackSet est une ressource régionale. Si vous créez un StackSet dans un Région AWS, vous ne pouvez le voir ou le modifier que lorsque vous visualisez cette région.

Modèles d'autorisation pour StackSets

Vous pouvez créer StackSets à l'aide d'autorisations autogérées ou d'autorisations gérées par le service.

Avec les autorisations autogérées, vous créez les rôles IAM nécessaires StackSets au déploiement sur plusieurs comptes et régions. Ces rôles sont nécessaires pour établir une relation de confiance entre le compte à StackSet partir duquel vous administrez et le compte sur lequel vous déployez des instances de stack. À l'aide de ce modèle d'autorisations, vous StackSets pouvez effectuer un déploiement sur tous ceux Compte AWS dans lesquels vous êtes autorisé à créer un rôle IAM.

Avec les autorisations gérées par le service vous pouvez déployer des instances de piles sur des comptes gérés par AWS Organizations. En utilisant ce modèle d'autorisations, vous n'avez pas à

créer les rôles IAM nécessaires ; il StackSets crée les rôles IAM en votre nom. Avec ce modèle, vous pouvez également activer les déploiements automatiques sur les comptes qui seront ajoutés ultérieurement à votre organisation.

AWS Organizations s'intègre à votre environnement CloudFormation et vous aide à le gérer et à le gouverner de manière centralisée à mesure que vous adaptez et développez vos AWS ressources.

- **Compte de gestion** : le compte que vous utilisez pour créer l'organisation. Pour plus d'informations, consultez [Terminologie et concepts pour AWS Organizations](#).
- **Administrateur délégué** : un AWS service compatible peut enregistrer un compte de AWS membre dans l'organisation en tant qu'administrateur des comptes de l'organisation dans ce service. Pour plus d'informations, consultez [AWS les services que vous pouvez utiliser avec AWS Organizations](#).

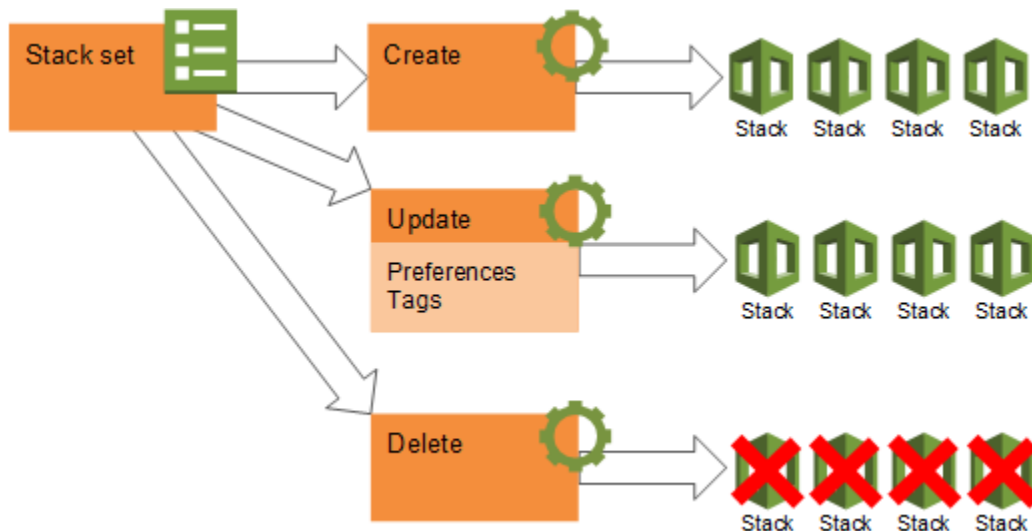
Pour plus d'informations sur la création et la gestion à l' StackSets aide d'autorisations gérées par le service, consultez les rubriques suivantes :

- [Activez un accès sécurisé pour StackSets avec AWS Organizations](#)
- [Enregistrer un compte membre d'administrateur délégué](#)
- [Créez CloudFormation StackSets avec des autorisations gérées par le service](#)

Instances de piles

Une instance de pile est une référence à une pile d'un compte de destination dans une région. Une instance de pile peut exister sans pile. Par exemple, si la pile n'a pas pu être créée pour une raison ou une autre, l'instance de pile indique le motif de l'échec de la création de la pile. Une instance de pile n'est associée qu'à une seule instance StackSet.

La figure suivante montre les relations logiques entre StackSets les opérations de pile et les piles. Lorsque vous mettez à jour un StackSet, toutes les instances de stack associées sont mises à jour dans tous les comptes et régions.



StackSet opérations

Vous pouvez effectuer les opérations suivantes sur StackSets.

Créez StackSet

La création d'un nouveau CloudFormation modèle StackSet implique de spécifier un modèle que vous souhaitez utiliser pour créer des piles, de spécifier les comptes cibles dans lesquels vous souhaitez créer des piles et d'identifier ceux Régions AWS dans lesquels vous souhaitez déployer des piles dans vos comptes cibles. A StackSet garantit le déploiement cohérent des mêmes ressources, avec les mêmes paramètres, vers tous les comptes cibles spécifiés dans les régions que vous choisissez.

Mettre à jour StackSet

Lorsque vous mettez à jour un StackSet, vous transférez les modifications vers les piles de votre StackSet. Vous pouvez mettre à jour un StackSet de l'une des manières suivantes. Les mises à jour de votre modèle concernent toujours toutes les piles ; vous ne pouvez pas mettre à jour le modèle de manière sélective pour certaines piles du StackSet, mais pas pour d'autres.

- Modifiez les paramètres existants dans le modèle ou ajoutez des ressources, par exemple mettez à jour les valeurs des paramètres pour un service spécifique, ou ajoutez des instances Amazon EC2.
- Remplacez le modèle par un autre modèle.
- Ajoutez des piles dans les comptes de destination existants ou supplémentaires appartenant aux régions existantes ou aux régions supplémentaires.

Supprimer des piles

Lorsque vous supprimez des piles, vous supprimez chaque pile et toutes les ressources qui lui sont associées des comptes de destination dans les régions que vous spécifiez. Pour supprimer des piles, vous pouvez procéder de différentes façons.

- Supprimer des piles dans certains comptes de destination tout en laissant les autres piles s'exécuter dans les autres comptes de destination.
- Supprimer des piles dans certaines régions tout en laissant les piles s'exécuter dans les autres régions.
- Supprimez les piles de votre ordinateur StackSet, mais enregistrez-les pour qu'elles continuent à fonctionner indépendamment de vous StackSet en choisissant l'option Retain Stacks. Vous pouvez ensuite gérer les piles conservées en dehors de votre entrée StackSet . CloudFormation
- Supprimez toutes les piles de votre ordinateur StackSet, afin de préparer la suppression complète StackSet de la vôtre.

Supprimer StackSet

Vous ne pouvez supprimer le vôtre StackSet que s'il ne contient aucune instance de pile.

StackSet options de fonctionnement

Les options décrites dans cette section permettent de contrôler le temps et le nombre d'échecs autorisés pour effectuer des StackSet opérations avec succès, et vous évitent de perdre des ressources de pile.

Nombre maximal de comptes en parallèle

Ce paramètre, disponible dans les flux de travail de création, de mise à jour et de suppression, permet de spécifier le nombre ou le pourcentage maximal de comptes cibles dans lequel une opération est effectuée simultanément. Un nombre ou pourcentage faible signifie qu'une opération est effectuée dans peu de comptes de destination à la fois. Les opérations sont effectuées dans une seule région à la fois, dans l'ordre spécifié dans la zone Deployment order (Ordre de déploiement). Par exemple, si vous déployez des piles dans 10 comptes cibles dans deux régions et que vous définissez Maximum concurrent accounts (Nombre maximal de comptes en parallèle) sur 50 et By percentage (Par pourcentage), le logiciel déploie les piles dans cinq comptes dans la première région, puis dans les cinq comptes suivants dans la première région, avant de passer à la région suivante et de lancer le déploiement dans les cinq premiers comptes cibles.

Lorsque vous choisissez Par pourcentage, si le pourcentage spécifié ne représente pas un nombre entier de vos comptes spécifiés, CloudFormation arrondit au chiffre inférieur. Par exemple, si vous déployez des piles sur 10 comptes cibles et que vous définissez le nombre maximal de comptes simultanés à 25 et par pourcentage, CloudFormation arrondit le nombre de 2,5 piles simultanément (ce qui ne serait pas possible) au déploiement de deux piles simultanément.

Notez que ce paramètre vous permet de spécifier la valeur maximale pour les opérations. Pour les déploiements importants, dans certaines circonstances, le nombre réel de comptes traités simultanément peut être inférieur en raison de la limitation de service.

Le nombre maximal de comptes simultanés peut dépendre de la valeur de la tolérance aux pannes en fonction de votre mode de simultanéité. Si votre mode de simultanéité est défini sur Tolérance stricte aux pannes, le nombre maximal de comptes simultanés peut être supérieur d'un plus un au paramètre de tolérance aux pannes.

Mode simultané

Ce paramètre, disponible dans les processus de création, de mise à jour et de suppression, vous permet de choisir le comportement du niveau de simultanéité pendant StackSet les opérations. Pour de plus amples informations, veuillez consulter [Choisissez le mode de simultanéité pour CloudFormation StackSets](#).

Tolérance aux pannes

Ce paramètre, disponible dans les flux de travail de création, de mise à jour et de suppression, vous permet de spécifier le nombre ou le pourcentage maximum d'échecs d'opérations de pile pouvant survenir, par région, au-delà duquel CloudFormation une opération est automatiquement interrompue. Si le nombre ou le pourcentage est faible, l'opération est effectuée sur peu de piles, mais vous pouvez commencer plus rapidement à résoudre les problèmes d'échec des opérations. Par exemple, si vous mettez à jour 10 piles dans 10 comptes de destination dans trois régions, et que vous définissez Failure tolerance (Tolérance aux pannes) sur 20 et By percentage (Par pourcentage), cela signifie que l'opération peut se poursuivre si le nombre d'échecs de mises à jour de pile ne dépasse pas deux échecs. Si une troisième pile dans la même région échoue, CloudFormation arrête l'opération. Si une pile n'a pas pu être mise à jour dans la première région, l'opération de mise à jour continue dans cette région, puis passe à la région suivante. Si deux piles ne peuvent pas être mises à jour dans la deuxième région, la tolérance d'échec atteint 20 % ; si une troisième pile de la région échoue, CloudFormation arrête l'opération de mise à jour et ne passe pas aux régions suivantes.

Lorsque vous choisissez Par pourcentage, si le pourcentage spécifié ne représente pas un nombre entier de vos cumuls dans chaque région, CloudFormation arrondissez au chiffre inférieur. Par exemple, si vous déployez des piles sur 10 comptes cibles dans trois régions et que vous définissez la tolérance d'échec à 25 et que vous CloudFormation arrondissez en pourcentage d'une tolérance d'échec de 2,5 piles (ce qui ne serait pas possible) à une tolérance d'échec de deux piles par région.

Conserver les piles

Ce paramètre, disponible dans les flux de travail de suppression de piles, vous permet de maintenir les piles et leurs ressources en activité même après avoir supprimé des piles d'un StackSet. Lorsque vous conservez les piles, CloudFormation laisse intactes les piles placées dans des comptes et régions individuels. Les piles se dissocient de la StackSet, mais économisent la pile et ses ressources. Une fois qu'une opération de suppression de piles est terminée, vous gérez les piles conservées dans CloudFormation le compte cible (et non le compte administrateur) qui a créé les piles.

Region concurrency (Simultanéité de région)

Ce paramètre, disponible dans les processus de création, de mise à jour et de suppression, vous permet de choisir le mode de StackSets déploiement dans les régions.

Séquentiel — Déployez l' StackSets opération dans une région à la fois, comme indiqué dans la case de commande Déploiement par région, à condition que les échecs de déploiement d'une région ne dépassent pas la tolérance d'échec spécifiée. Le déploiement séquentiel est la sélection par défaut.

Parallèle — Déployez les StackSets opérations simultanément dans toutes les régions spécifiées, à condition que les échecs de déploiement d'une région ne dépassent pas la tolérance d'échec spécifiée.

Étiquettes

Vous pouvez ajouter des balises lors des opérations de StackSet création et de mise à jour en spécifiant des paires clé/valeur. Les balises sont utiles pour trier et filtrer les StackSet ressources à des fins de facturation et de répartition des coûts. Pour plus d'informations sur l'utilisation des balises AWS, consultez la section [Organisation et suivi des coûts à l'aide des balises de répartition des AWS coûts](#) dans le Guide de AWS Billing and Cost Management l'utilisateur. Une fois que vous avez spécifié la paire clé-valeur, choisissez + pour enregistrer la balise. Vous pouvez supprimer des balises que vous n'utilisez plus en sélectionnant le X rouge à droite d'une balise.

Des balises que vous StackSets appliquez à toutes les piles et aux ressources créées par vos piles. Vous pouvez également ajouter des balises au niveau de la pile uniquement dans CloudFormation, mais il est possible que ces balises n'apparaissent pas dans. StackSets

Bien qu' StackSets il n'ajoute aucune balise définie par le système, vous ne devez pas commencer le nom des clés d'aucune balise par la chaîne. aws :

StackSets codes de statut

CloudFormation StackSets génère des codes d'état pour les StackSet opérations.

Le tableau suivant décrit les codes d'état des StackSet opérations.

RUNNING

L'opération est actuellement en cours.

SUCCEEDED

L'opération s'est terminée sans dépasser la tolérance aux pannes autorisée pour l'opération.

FAILED

Le nombre de piles sur lesquelles l'opération a échoué a dépassé le seuil de tolérance aux pannes défini par l'utilisateur. La valeur de tolérance aux pannes que vous avez définie pour une opération est appliquée pour chaque région lors de la création et de la mise à jour des piles. Si le nombre de piles en échec dans une région dépasse la valeur de tolérance aux pannes, l'état de l'opération dans la région passe à FAILED. Le statut de l'opération dans son ensemble est également défini sur FAILED, et CloudFormation annule l'opération dans toutes les régions restantes.

QUEUED

[Service-managed permissions] Pour les déploiements automatiques nécessitant une séquence d'opérations, l'opération est mise en file d'attente afin d'être exécutée. Par exemple :

- Le déplacement d'un compte d'une unité organisationnelle (OU1UO) à une autre déclenche un déploiement automatique. OU2 StackSets exécute une opération de suppression pour supprimer l'instance de pile du OU1 compte cible dans la région cible et met en file d'attente une opération de création pour ajouter une instance de pile au OU2 compte cible dans la région cible.
- L'ajout d'un compte AccountA à une unité d'organisation déclenche un déploiement automatique. StackSets exécute une opération de création pour ajouter une instance de

pile AccountA dans la région cible. Si vous ajoutez un autre compte AccountB à l'unité d'organisation alors que cette opération de création est en cours d'exécution, une deuxième opération de création est mise en StackSets file d'attente. Lorsque la première opération de création est terminée, StackSets exécute la deuxième opération de création pour ajouter une instance de pile AccountB dans la région cible.

STOPPING

L'opération est en cours d'arrêt, à la demande de l'utilisateur.

STOPPED

L'opération s'est arrêtée à la demande de l'utilisateur.

Codes d'état de l'instance de pile

CloudFormation StackSets génère des codes d'état pour les instances de pile.

Le tableau suivant décrit les codes d'état pour les instances de pile qu'ils contiennent StackSets.

CURRENT

La pile est à jour avec le StackSet.

OUTDATED

La pile n'est pas à jour avec le StackSet pour l'une des raisons suivantes.

- Une opération [CreateStackSet](#) ou [UpdateStackSet](#) sur la pile associée a échoué.
- La pile était impliquée dans une opération [CreateStackSet](#) ou [UpdateStackSet](#) qui a échoué ou a été arrêtée avant la création ou la mise à jour de la pile.

INOPERABLE

Une opération [DeleteStackInstances](#) a échoué et a laissé la pile dans un état instable. Les piles affichant cet état sont exclues des opérations [UpdateStackSet](#) suivantes. Vous devrez peut-être effectuer une opération [DeleteStackInstances](#), avec `RetainStacks` défini sur `true`, pour supprimer l'instance de pile, puis supprimer la pile manuellement.

CANCELLED

L'opération dans le compte et la région spécifiés a été annulée. Cela se produit parce qu'un utilisateur a arrêté l' StackSet opération ou parce que les StackSet opérations dépassent la tolérance d'échec.

FAILED

Échec de l'opération dans le compte et la région spécifiés. Si l' StackSet opération échoue sur un nombre suffisant de comptes au sein d'une région, la tolérance d'échec pour l' StackSet opération dans son ensemble risque d'être dépassée.

FAILED_IMPORT

L'importation de l'instance de pile dans le compte et la région spécifiés a échoué et a laissé la pile dans un état instable. Une fois les problèmes à l'origine de l'échec résolus, l'opération d'importation peut être réessayée. Si un nombre suffisant d' StackSet opérations échouent sur un nombre suffisant de comptes au sein d'une région, la tolérance d'échec pour l' StackSet opération dans son ensemble risque d'être dépassée.

PENDING

L'opération dans le compte et la région spécifiés n'a pas encore démarré.

RUNNING

L'opération dans le compte et la région spécifiés est en cours.

SKIPPED_SUSPENDED_ACCOUNT

L'opération dans le compte et la région spécifiés a été ignorée, car le compte était suspendu au moment de l'opération.

SUCCEEDED

L'opération dans le compte et la région spécifiés s'est déroulée avec succès.

Conditions préalables à l'utilisation CloudFormation StackSets

StackSets étendez les fonctionnalités des piles afin de pouvoir créer, mettre à jour ou supprimer des piles sur plusieurs comptes et régions en une seule opération.

Comme vous pouvez StackSets effectuer des opérations de stack sur plusieurs comptes, avant de pouvoir créer le premier, StackSet vous devez disposer des autorisations nécessaires définies dans votre Comptes AWS.

Vous pouvez gérer à StackSets l'aide d'autorisations autogérées ou gérées par le service.

- Pour l'autogestion StackSets, vous devez créer et gérer des rôles IAM dans chaque compte cible et. Région AWS Pour de plus amples informations, veuillez consulter [Accorder des autorisations autogérées](#).
- Pour les services gérés StackSets, vous n'avez pas besoin de créer et de gérer manuellement les rôles IAM dans chaque compte ; il AWS s'occupe de la création des rôles et des autorisations pour vous. Pour de plus amples informations, veuillez consulter [Activer l'accès approuvé](#).

Rubriques

- [Préparez-vous à effectuer StackSet Régions AWS des opérations désactivées par défaut](#)
- [Accorder des autorisations autogérées](#)
- [Activez un accès sécurisé pour StackSets avec AWS Organizations](#)

Préparez-vous à effectuer StackSet Régions AWS des opérations désactivées par défaut

Régions AWS introduites après le 20 mars 2019, telles que la région Asie-Pacifique (Hong Kong), sont désactivées par défaut. Vous devez activer ces régions pour votre ou vos comptes avant de pouvoir les utiliser. Pour plus d'informations, consultez la section [Activer ou désactiver Régions AWS dans votre compte](#) dans le Guide de référence Gestion de compte AWS .

Pour créer un StackSet compte administrateur (si vous utilisez des autorisations autogérées) ou un StackSet compte de gestion de l'organisation (si vous utilisez des autorisations gérées par le service) dans une région désactivée par défaut, vous devez d'abord activer cette région pour le compte administrateur ou de gestion.

Pour CloudFormation créer ou mettre à jour correctement une instance de stack :

- le compte cible doit résider dans une région actuellement activée pour ce compte cible.
- Le StackSet compte administrateur ou le compte de gestion de l'organisation doit avoir la même région activée que le compte cible.

Important

Sachez que pendant les StackSet opérations, l'administrateur et les comptes cibles échangent des métadonnées concernant les comptes eux-mêmes, en plus des StackSet instances StackSet et des instances impliquées.

En outre, si vous désactivez une région qui contient un compte dans lequel résident StackSet des instances, vous êtes responsable de la suppression de ces instances ou ressources, si vous le souhaitez. En outre, sachez que les métadonnées concernant le compte cible dans la région désactivée seront conservées dans le compte administrateur.

Accorder des autorisations autogérées

Cette rubrique fournit des instructions sur la façon de créer les rôles de service IAM nécessaires StackSets au déploiement sur plusieurs comptes et Régions AWS avec des autorisations autogérées. Ces rôles sont nécessaires pour établir une relation de confiance entre le compte à StackSet partir duquel vous administrez et le compte sur lequel vous déployez des instances de stack. À l'aide de ce modèle d'autorisations, vous StackSets pouvez effectuer un déploiement sur tous ceux Compte AWS dans lesquels vous êtes autorisé à créer un rôle IAM.

Pour utiliser des autorisations gérées par le service, consultez [Activer l'accès approuvé](#) à la place.

Rubriques

- [Présentation des autorisations autogérées](#)
- [Autorisation accordée à tous les utilisateurs du compte administrateur de gérer les piles dans tous les comptes cibles](#)
- [Configurer des options d'autorisation avancées pour les StackSet opérations](#)
- [Configurez des clés globales pour atténuer les problèmes de député confus](#)

Présentation des autorisations autogérées

Avant de créer un StackSet avec des autorisations autogérées, vous devez avoir créé des rôles de service IAM dans chaque compte.

Les étapes de base sont les suivantes :

1. Déterminez quel Compte AWS est le compte administrateur.

StackSets sont créés dans ce compte administrateur. Un compte cible est le compte dans lequel vous créez des piles individuelles appartenant à un StackSet.

2. Déterminez comment vous souhaitez structurer les autorisations pour StackSet.

La configuration d'autorisations la plus simple (et la plus permissive) consiste à donner à tous les utilisateurs et groupes du compte administrateur la possibilité de créer et de mettre à jour tous les droits StackSets gérés via ce compte. Si vous avez besoin d'un contrôle plus fin, vous pouvez configurer des autorisations pour spécifier :

- Quels utilisateurs et groupes peuvent effectuer StackSet des opérations sur quels comptes cibles.
- Quelles ressources les utilisateurs et les groupes peuvent inclure dans leurs StackSets.
- Quelles StackSet opérations peuvent être effectuées par des utilisateurs et des groupes spécifiques.

3. Créez les rôles de service IAM nécessaires dans vos comptes d'administrateur et de destination pour définir les autorisations souhaitées.

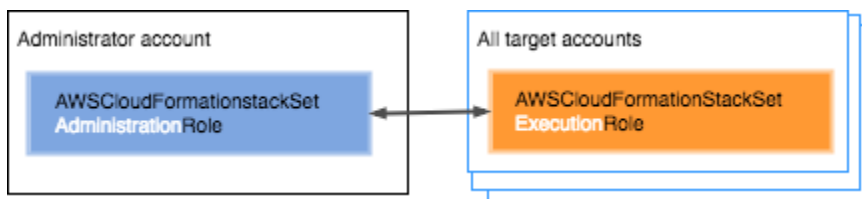
Plus précisément, les deux rôles requis sont les suivants :

- `AWSCloudFormationStackSetAdministrationRole`— Ce rôle est déployé sur le compte administrateur.
- `AWSCloudFormationStackSetExecutionRole`— Ce rôle est déployé sur tous les comptes sur lesquels vous créez des instances de stack.

Autorisation accordée à tous les utilisateurs du compte administrateur de gérer les piles dans tous les comptes cibles

Cette section explique comment configurer des autorisations pour permettre à tous les utilisateurs et groupes du compte administrateur d'effectuer StackSet des opérations sur tous les comptes cibles. Elle vous guide dans la création des rôles de service IAM requis dans vos comptes administrateur et cible. Toute personne disposant d'un compte administrateur peut alors créer, mettre à jour ou supprimer n'importe quelle pile dans n'importe quel compte cible.

En structurant les autorisations de cette manière, les utilisateurs ne transmettent aucun rôle d'administration lors de la création ou de la mise à jour d'un StackSet.



Administrator account

Dans le compte administrateur, créez un rôle IAM nommé `AWSCloudFormationStackSetAdministrationRole`.

Vous pouvez le faire en créant une pile à partir du CloudFormation modèle disponible dans <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.yml>.

Exemple Exemple de politique d'autorisations

Le rôle d'administration créé par le modèle précédent inclut la politique d'autorisation suivante.

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

Exemple Exemple de politique d'approbation 1

Le modèle précédent inclut également la politique d'approbation suivante qui accorde au service l'autorisation d'utiliser le rôle d'administration et les autorisations associées à ce rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      }
    }
  ]
}
```

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

Exemple Exemple de politique d'approbation 2

Pour déployer des instances de pile dans un compte cible situé dans une région désactivée par défaut, vous devez également inclure le principal régional pour cette région. Chaque région désactivée par défaut aura son propre principal de service régional.

L'exemple de politique d'approbation suivante accorde au service l'autorisation d'utiliser le rôle d'administration dans la région Asie-Pacifique (Hong Kong) (ap-east-1), une région désactivée par défaut.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": [  
          "cloudformation.amazonaws.com",  
          "cloudformation.ap-east-1.amazonaws.com"  
        ]  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

Pour de plus amples informations, veuillez consulter [Préparez-vous à effectuer StackSet Régions AWS des opérations désactivées par défaut](#). Pour obtenir une liste des codes de région, consultez [Points de terminaison régionaux](#) dans le Guide Références générales AWS .

Target accounts

Dans chaque compte cible, créez un rôle de service nommé `AWSCloudFormationStackSetExecutionRole` qui approuve le compte administrateur. Ce rôle doit

avoir ce nom exact. Vous pouvez le faire en créant une pile à partir du CloudFormation modèle disponible dans <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.yml>. Lorsque vous utilisez ce modèle, vous êtes invité à fournir l'ID de compte administrateur avec lequel votre compte cible doit avoir une relation d'approbation.

 Important

Sachez que ce modèle accorde l'accès administrateur. Après avoir utilisé le modèle pour créer un rôle d'exécution de compte cible, vous devez étendre les autorisations de la déclaration de politique aux types de ressources que vous créez en utilisant StackSets.

Le rôle de service du compte cible nécessite des autorisations pour effectuer toutes les opérations spécifiées dans votre CloudFormation modèle. Par exemple, si votre modèle crée un compartiment S3, vous avez besoin d'autorisations pour créer des objets pour S3. Votre compte de destination doit toujours disposer des autorisations CloudFormation complètes, notamment des autorisations de création, de mise à jour, de suppression et de description des piles.

Exemple Exemple de politique d'autorisations 1

Le rôle créé par ce modèle active la politique suivante sur votre compte de destination.

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

Exemple Exemple de politique d'autorisations 2

L'exemple suivant montre une déclaration de politique avec les autorisations minimales StackSets pour fonctionner. Pour créer des piles dans des comptes cibles qui utilisent des ressources

provenant de services autres que CloudFormation, vous devez ajouter ces actions de service et ces ressources à la déclaration de `AWS::CloudFormation::StackSetExecutionRolePolicy` de chaque compte cible.

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

Exemple Exemple de politique d'approbation

La relation d'approbation suivante est créée par le modèle. L'identifiant du compte administrateur est affiché sous la forme `admin_account_id`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Vous pouvez configurer la relation d'approbation du rôle d'exécution d'un compte cible existant pour approuver un rôle spécifique dans le compte administrateur. Si vous supprimez le rôle dans le compte administrateur et que vous en créez un nouveau pour le remplacer, vous devez

configurer les relations de confiance de votre compte cible avec le nouveau rôle du compte administrateur, représenté *admin_account_id* dans l'exemple précédent.

Configurer des options d'autorisation avancées pour les StackSet opérations

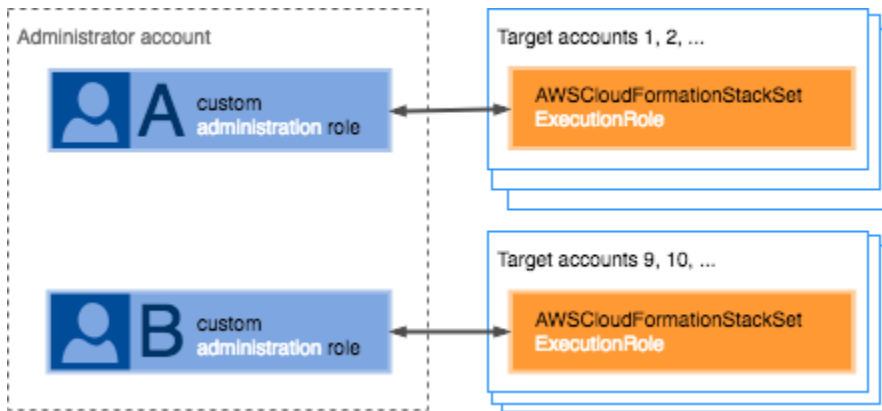
Si vous avez besoin d'un contrôle plus précis sur StackSets ce que les utilisateurs et les groupes créent par le biais d'un seul compte administrateur, vous pouvez utiliser les rôles IAM pour spécifier :

- Quels utilisateurs et groupes peuvent effectuer StackSet des opérations sur quels comptes cibles.
- Quelles ressources les utilisateurs et les groupes peuvent inclure dans leurs StackSets.
- Quelles StackSet opérations peuvent être effectuées par des utilisateurs et des groupes spécifiques.

Contrôlez quels utilisateurs peuvent effectuer StackSet des opérations sur des comptes cibles spécifiques

Utilisez des rôles d'administration personnalisés pour contrôler quels utilisateurs et groupes peuvent effectuer StackSet des opérations sur quels comptes cibles. Vous souhaitez peut-être contrôler quels utilisateurs du compte administrateur peuvent effectuer des StackSet opérations sur quels comptes cibles. Pour ce faire, vous créez une relation de confiance entre chaque compte cible et un rôle d'administration personnalisé spécifique, plutôt que de créer le rôle de `AWSCloudFormationStackSetAdministrationRoleservice` dans le compte administrateur lui-même. Vous activez ensuite des utilisateurs et des groupes spécifiques pour qu'ils utilisent le rôle d'administration personnalisé lors de l'exécution d' StackSet opérations sur un compte cible spécifique.

Par exemple, vous pouvez créer un rôle A et un rôle B dans votre compte d'administrateur. Vous pouvez ensuite accorder au rôle A l'autorisation d'accéder au compte de destination 1 via le compte 8. Vous pouvez enfin accorder au rôle B l'autorisation d'accéder au compte de destination 9 via le compte 16.



La configuration des autorisations nécessaires implique de définir un rôle d'administration personnalisé, de créer un rôle de service pour le compte cible et d'accorder aux utilisateurs l'autorisation de transmettre le rôle d'administration personnalisé lors de l'exécution StackSet des opérations.

En général, voici comment cela fonctionne une fois que vous avez les autorisations nécessaires en place : lors de la création d'un StackSet, l'utilisateur doit spécifier un rôle d'administration personnalisé. L'utilisateur doit disposer de l'autorisation nécessaire pour transmettre le rôle à CloudFormation. En outre, le rôle d'administration personnalisé doit avoir une relation de confiance avec les comptes cibles spécifiés pour le StackSet. CloudFormation crée le rôle d'administration personnalisé StackSet et y associe le rôle d'administration personnalisé. Lors de la mise à jour d'un StackSet, l'utilisateur doit spécifier explicitement un rôle d'administration personnalisé, même s'il s'agit du même rôle d'administration personnalisé que celui utilisé StackSet précédemment. CloudFormation utilise ce rôle pour mettre à jour la pile, sous réserve des exigences ci-dessus.

Administrator account

Exemple Exemple de politique d'autorisations

Pour chacun StackSet, créez un rôle d'administration personnalisé avec les autorisations nécessaires pour assumer le rôle d'exécution du compte cible.

Le nom du rôle d'exécution du compte cible doit être le même dans tous les comptes cibles. Si le nom du rôle est `AWSCloudFormationStackSetExecutionRole`, le StackSets utilise automatiquement lors de la création d'un StackSet. Si vous spécifiez un nom de rôle personnalisé, les utilisateurs doivent fournir le nom du rôle d'exécution lors de la création d'un StackSet.

Créez un [rôle de service IAM](#) avec un nom personnalisé et la politique d'autorisations suivante. Dans les exemples ci-dessous, `custom_execution_role` fait référence au rôle d'exécution dans les comptes cibles.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/custom_execution_role"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Pour spécifier plusieurs comptes dans une seule instruction, séparez-les par des virgules.

```
"Resource": [
  "arn:aws:iam::target_account_id_1:role/custom_execution_role",
  "arn:aws:iam::target_account_id_2:role/custom_execution_role"
]
```

Vous pouvez spécifier tous les comptes cibles en utilisant un caractère générique (*) à la place d'un ID de compte.

```
"Resource": [
  "arn:aws:iam::*:role/custom_execution_role"
]
```

Exemple Exemple de politique d'approbation 1

Vous devez fournir une politique d'approbation pour le rôle de service afin de définir quels principaux IAM peuvent assumer le rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Effect": "Allow",
    "Principal": {
        "Service": "cloudformation.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
]
```

Exemple Exemple de politique d'approbation 2

Pour déployer des instances de pile dans un compte cible situé dans une région désactivée par défaut, vous devez également inclure le principal régional pour cette région. Chaque région désactivée par défaut aura son propre principal de service régional.

L'exemple de politique d'approbation suivante accorde au service l'autorisation d'utiliser le rôle d'administration dans la région Asie-Pacifique (Hong Kong) (ap-east-1), une région désactivée par défaut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pour de plus amples informations, veuillez consulter [Préparez-vous à effectuer StackSet Régions AWS des opérations désactivées par défaut](#). Pour obtenir la liste des codes de région, consultez [Points de terminaison régionaux](#) dans le Guide de référence générale AWS .

Exemple Exemple de politique de transfert de rôle

Vous avez également besoin d'une politique d'autorisations IAM pour vos utilisateurs IAM qui permet à l'utilisateur de transmettre le rôle d'administration personnalisé lors de l'exécution StackSet d'opérations. Pour plus d'informations, consultez la section [Octroi d'autorisations à un utilisateur pour transférer un rôle à un service AWS](#).

Dans l'exemple ci-dessous, *customized_admin_role* fait référence au rôle d'administration que l'utilisateur doit transmettre.

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

Target accounts

Dans chaque compte cible, créez un rôle de service qui fait confiance au rôle d'administration personnalisé que vous voulez utiliser avec ce compte.

Le rôle de compte cible nécessite des autorisations pour effectuer toutes les opérations spécifiées dans votre CloudFormation modèle. Par exemple, si votre modèle crée un compartiment S3, vous devez disposer des autorisations permettant de créer des objets dans S3. Votre compte cible a toujours besoin d' CloudFormation autorisations complètes, notamment des autorisations pour créer, mettre à jour, supprimer et décrire des piles.

Le nom du rôle du compte cible doit être le même dans tous les comptes cibles. Si le nom du rôle est `AWSCloudFormationStackSetExecutionRole`, l' `StackSets` utilise automatiquement lors de la création d'un StackSet. Si vous spécifiez un nom de rôle personnalisé, les utilisateurs doivent fournir le nom du rôle d'exécution lors de la création d'un StackSet.

Exemple Exemple de politique d'autorisations

L'exemple suivant montre une déclaration de politique avec les autorisations minimales StackSets pour fonctionner. Pour créer des piles dans des comptes cibles qui utilisent des ressources provenant de services autres que CloudFormation, vous devez ajouter ces actions de service et ces ressources à la politique d'autorisation.

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

Exemple Exemple de politique d'approbation

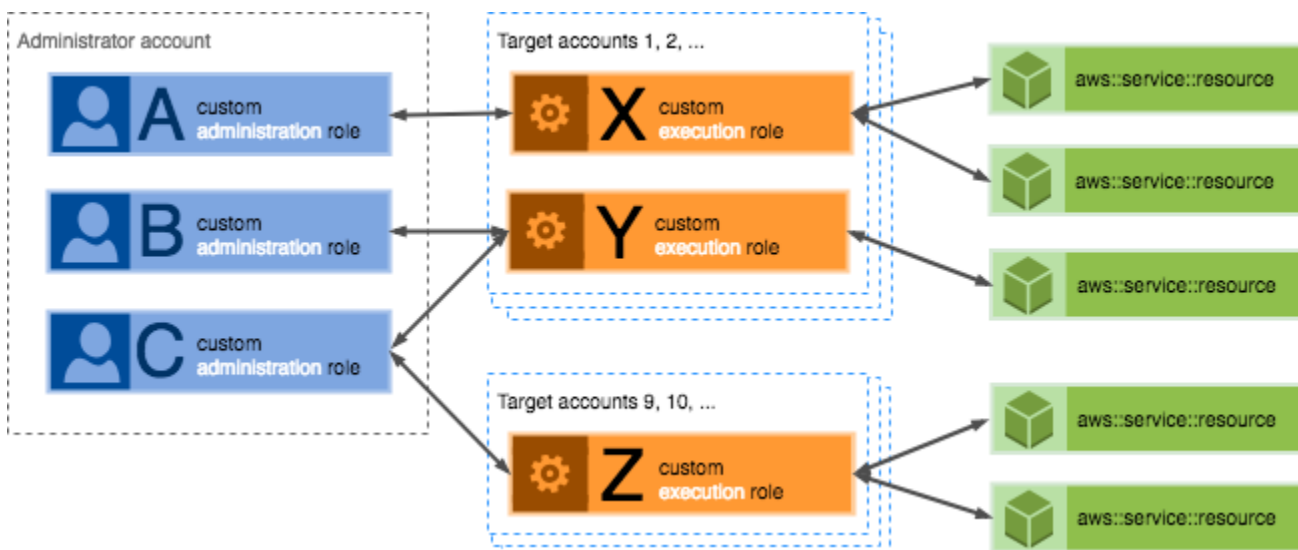
Vous devez fournir la politique d'approbation suivante lorsque vous créez le rôle pour définir la relation d'approbation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Contrôlez les ressources que les utilisateurs peuvent inclure dans des StackSets

Utilisez des rôles d'exécution personnalisés pour contrôler les ressources de pile que les utilisateurs et les groupes peuvent inclure dans leur pile StackSets. Par exemple, vous souhaitez peut-être configurer un groupe qui ne peut inclure que des ressources liées à Amazon S3 dans ce qu'il crée, tandis StackSets qu'une autre équipe ne peut inclure que des ressources DynamoDB. Pour ce faire, vous créez une relation d'approbation entre le rôle d'administrateur personnalisé pour chaque groupe et un rôle d'exécution personnalisé pour chaque jeu de ressources. Le rôle d'exécution personnalisé définit les ressources de pile qui peuvent être incluses StackSets. Le rôle d'administration personnalisé réside dans le compte administrateur, tandis que le rôle d'exécution personnalisé réside dans chaque compte cible dans lequel vous souhaitez créer à StackSets l'aide des ressources définies. Vous activez ensuite des utilisateurs et des groupes spécifiques pour qu'ils utilisent le rôle d'administration personnalisé lors de l'exécution StackSets des opérations.

Par exemple, vous pouvez créer des rôles d'administration personnalisés A, B et C dans le compte administrateur. Les utilisateurs et les groupes autorisés à utiliser le rôle A peuvent créer StackSets des ressources contenant les ressources de pile spécifiquement répertoriées dans le rôle d'exécution personnalisé X, mais pas celles des rôles Y ou Z, ou les ressources non incluses dans un rôle d'exécution.



Lors de la mise à jour d'un StackSet, l'utilisateur doit spécifier explicitement un rôle d'administration personnalisé, même s'il s'agit du même rôle d'administration personnalisé que celui utilisé StackSet précédemment. CloudFormation effectue la mise à jour en utilisant le rôle d'administration personnalisé spécifié, à condition que l'utilisateur soit autorisé à effectuer des opérations sur ce rôle StackSet.

De même, l'utilisateur peut également spécifier un rôle d'exécution personnalisé. S'ils spécifient un rôle d'exécution personnalisé, CloudFormation utilise ce rôle pour mettre à jour la pile, sous réserve des exigences ci-dessus. Si l'utilisateur ne spécifie pas de rôle d'exécution personnalisé, CloudFormation effectue la mise à jour en utilisant le rôle d'exécution personnalisé précédemment associé au StackSet, à condition que l'utilisateur soit autorisé à effectuer des opérations sur ce rôle StackSet.

Administrator account

Créez un rôle d'administration personnalisé dans votre compte administrateur, comme indiqué dans [Contrôlez quels utilisateurs peuvent effectuer StackSet des opérations sur des comptes cibles spécifiques](#). Incluez une relation d'approbation entre le rôle d'administration personnalisé et les rôles d'exécution personnalisés que vous voulez utiliser.

Exemple Exemple de politique d'autorisations

L'exemple suivant est une politique d'autorisation à la fois pour AWSCloudFormationStackSetExecutionRolecelle définie pour le compte cible, en plus d'un rôle d'exécution personnalisé.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487980684000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole",
        "arn:aws:iam::*:role/custom_execution_role"
      ]
    }
  ]
}
```

Target accounts

Dans les comptes cibles dans lesquels vous souhaitez créer votre StackSets, créez un rôle d'exécution personnalisé qui accorde des autorisations aux services et aux ressources que vous souhaitez que les utilisateurs et les groupes puissent inclure dans StackSets.

Exemple Exemple de politique d'autorisations

L'exemple suivant fournit les autorisations minimales pour StackSets, ainsi que l'autorisation de créer des tables Amazon DynamoDB.

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

Exemple Exemple de politique d'approbation

Vous devez fournir la politique d'approbation suivante lorsque vous créez le rôle pour définir la relation d'approbation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
        },
        "Action": "sts:AssumeRole"
    }
}
```

Configurer des autorisations pour des StackSet opérations spécifiques

En outre, vous pouvez définir des autorisations pour lesquelles les utilisateurs et les groupes peuvent effectuer des StackSet opérations spécifiques, telles que la création, la mise à jour, la suppression StackSets ou le cumul d'instances. Pour plus d'informations, consultez la rubrique [Actions, ressources et clés de condition pour CloudFormation](#) dans la section Référence de l'autorisation de service.

Configurez des clés globales pour atténuer les problèmes de député confus

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui CloudFormation StackSets accordent un autre service à la ressource. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de stratégie.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez

la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:cloudformation::123456789012:*`. Dans la mesure du possible, utilisez `aws:SourceArn`, car il est plus spécifique. Utilisez `aws:SourceAccount` uniquement lorsque vous ne pouvez pas déterminer l'ARN ou le modèle d'ARN correct.

Lorsque StackSets assume le rôle d'administration dans votre compte d'administrateur, StackSets renseigne votre ID de compte administrateur et le nom de ressource StackSets Amazon (ARN). Vous pouvez donc définir des conditions pour les [clés globales](#) `aws:SourceAccount` et `aws:SourceArn` dans les relations de confiance afin d'éviter les [problèmes de député confus](#). L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale StackSets pour éviter le problème de confusion des adjoints.

Administrator account

Exemple Clés globales pour `aws:SourceAccount` et `aws:SourceArn`

Lors de l'utilisation StackSets, définissez les clés globales `aws:SourceAccount` et `aws:SourceArn` définissez votre politique de `AWSCloudFormationStackSetAdministrationRole` confiance pour éviter tout problème de confusion avec les adjoints.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
        }
      }
    }
  ]
}
```

```
}
```

Exemple StackSets ARNs

Spécifiez votre associé StackSets ARNs pour un contrôle plus précis.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": [
            "arn:aws:cloudformation:STACKSETS-REGION:111122223333:stackset/STACK-SET-ID-1",
            "arn:aws:cloudformation:STACKSETS-REGION:111122223333:stackset/STACK-SET-ID-2"
          ]
        }
      }
    }
  ]
}
```

Activez un accès sécurisé pour StackSets avec AWS Organizations

Cette rubrique fournit des instructions sur la façon d'activer l'accès sécurisé avec AWS Organizations, qui est requis pour le déploiement sur plusieurs comptes et StackSets à Régions AWS l'aide d'autorisations gérées par le service. Pour utiliser des autorisations autogérées, consultez [Accorder des autorisations autogérées](#) à la place.

Avant de créer un StackSet avec des autorisations gérées par le service, vous devez d'abord effectuer les tâches suivantes :

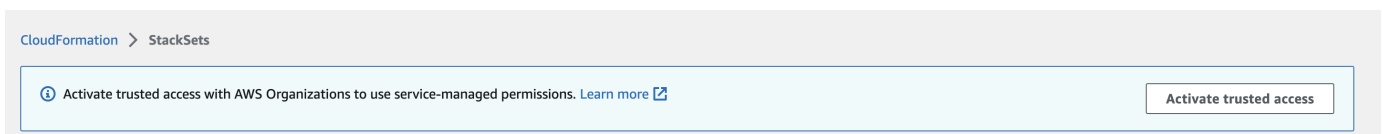
- [Activez toutes les fonctionnalités](#) dans AWS Organizations. Lorsque seules les fonctionnalités de facturation consolidée sont activées, vous ne pouvez pas créer un StackSet avec des autorisations gérées par le service.
- Activez un accès sécurisé avec AWS Organizations. Cette action permet de CloudFormation créer un rôle lié à un service dans le compte de gestion. Une fois l'accès sécurisé activé, lorsque vous créez un StackSet avec des autorisations gérées par le service, vous CloudFormation créez à la fois le rôle lié au service nécessaire et un rôle de service nommé `stacksets-exec-*` dans les comptes cibles (membres).

Lorsque l'accès sécurisé est activé, le compte de gestion et les comptes d'administrateur délégué peuvent créer et gérer des services gérés StackSets pour leur organisation.

Pour activer l'accès approuvé, vous devez être un utilisateur administrateur dans le compte de gestion. Un utilisateur administrateur est un utilisateur disposant de toutes les autorisations pour votre Compte AWS. Pour plus d'informations, consultez [Créer un utilisateur administrateur](#) dans le Guide de référence Gestion de compte AWS . Pour obtenir des recommandations sur la protection de la sécurité du compte de gestion, consultez [Bonnes pratiques pour le compte de gestion](#) dans le Guide de l'utilisateur AWS Organizations .

Pour activer l'accès sécurisé

1. Connectez-vous en AWS tant qu'administrateur du compte de gestion et ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez StackSets. Si l'accès sécurisé est désactivé, une bannière s'affiche pour vous inviter à activer l'accès sécurisé.



3. Choisissez Activer l'accès sécurisé.

L'accès sécurisé est activé avec succès lorsque la bannière suivante s'affiche.



 Note

Activate Organizations Access est identique à Enable Organizations Access, et Deactivate Organizations Access est identique à Disable Organizations Access. Ces conditions ont été mises à jour en fonction des directives de commercialisation.

Pour désactiver l'accès sécurisé

Voir [CloudFormation StackSets et AWS Organizations](#) dans le guide de AWS Organizations l'utilisateur.

Avant de pouvoir désactiver l'accès sécurisé avec AWS Organizations, vous devez désenregistrer tous les administrateurs délégués. Pour de plus amples informations, veuillez consulter [Enregistrer un administrateur délégué](#).

 Note

Pour plus d'informations sur l'utilisation des opérations API à la place de la console pour activer ou désactiver l'accès approuvé, veuillez consulter :

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

Rôles liés à un service

Le compte de gestion utilise le rôle `AWSServiceRoleForCloudFormationStackSetsOrgAdmin` lié au service. Vous ne pouvez modifier ou supprimer ce rôle que si l'accès sécurisé avec AWS Organizations est désactivé.

Chaque compte cible utilise un rôle `AWSServiceRoleForCloudFormationStackSetsOrgMember` lié à un service. Vous pouvez modifier ou supprimer ce rôle uniquement sous deux conditions : si l'accès sécurisé avec AWS Organizations est désactivé ou si le compte est supprimé de l'organisation ou de l'unité organisationnelle (UO) cible.

Pour plus d'informations, voir [CloudFormation StackSets et AWS Organizations](#) dans le guide de AWS Organizations l'utilisateur.

Enregistrer un compte membre d'administrateur délégué

Outre le compte de gestion de votre organisation, les comptes membres dotés d'autorisations d'administrateur déléguées peuvent être créés et gérés à l'aide d'autorisations gérées par les services pour l'organisation. StackSets avec des autorisations gérées par le service sont créés dans le compte de gestion, y compris StackSets par des administrateurs délégués. Pour être enregistré en tant qu'administrateur délégué de votre organisation, votre compte membre doit se trouver dans l'organisation. Pour plus d'informations sur l'adhésion à une organisation, voir [Inviter un Compte AWS homme à rejoindre votre organisation](#).

Jusqu'à cinq administrateurs délégués peuvent être enregistrés en même temps pour votre organisation. Les administrateurs délégués peuvent choisir de déployer sur tous les comptes de votre organisation ou sur des comptes spécifiques OUs. L'accès sécurisé AWS Organizations doit être activé avant que les administrateurs délégués puissent effectuer un déploiement sur des comptes gérés par des Organizations. Pour de plus amples informations, veuillez consulter [Activez un accès sécurisé pour StackSets avec AWS Organizations](#).

Important

Veuillez tenir compte des points suivants :

- Les administrateurs délégués disposent des autorisations complètes pour un déploiement dans les comptes de votre organisation. Le compte de gestion ne peut pas limiter les autorisations d'administrateur déléguées pour effectuer des déploiements OUs ou des StackSet opérations spécifiques.
- Veuillez vous assurer que vos administrateurs délégués disposent des autorisations `organizations:ListDelegatedAdministrators` afin d'éviter toute erreur potentielle.

Vous pouvez enregistrer des administrateurs délégués pour votre organisation dans les régions suivantes : USA Est (Ohio), USA Est (Virginie du Nord), USA Ouest (Californie du Nord), USA Ouest (Oregon), Asie-Pacifique (Mumbai), Asie-Pacifique (Séoul), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Canada (centre), Europe (Francfort), Europe (Irlande), Europe (Londres), Europe (Paris), Europe (Stockholm), Israël (Tel Aviv), Amérique du Sud (São Paulo), (US-Est) AWS GovCloud et (US-Ouest). AWS GovCloud

Vous pouvez enregistrer et désenregistrer les administrateurs délégués à l'aide de la [CloudFormation console AWS CLI](#), ou. [AWS SDKs](#)

Pour enregistrer un administrateur délégué (console)

1. Connectez-vous en AWS tant qu'administrateur du compte de gestion et ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation/>.
2. Dans le volet de navigation, choisissez StackSets.
3. Sous Delegated administrators (Administrateurs délégués), choisissez Register delegated administrator (Enregistrer l'administrateur délégué).
4. Dans la boîte de dialogue Register delegated administrator (Enregistrer l'administrateur délégué), choisissez Register delegated administrator (Enregistrer l'administrateur délégué).

Le message de réussite indique que le compte membre a bien été enregistré en tant qu'administrateur délégué.

Pour annuler l'enregistrement d'un administrateur délégué (console)

1. Connectez-vous en AWS tant qu'administrateur du compte de gestion et ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/>.
2. Dans le volet de navigation, choisissez StackSets.
3. Sous Delegated administrators (Administrateurs délégués), sélectionnez le compte dont vous souhaitez annuler l'enregistrement, puis choisissez Deregister (Annuler l'enregistrement).

Le message de réussite indique que l'enregistrement de ce compte membre en tant qu'administrateur délégué a bien été annulé.

Vous pouvez enregistrer à nouveau ce compte à tout moment.

Pour enregistrer un administrateur délégué (AWS CLI)

1. Ouvrez le AWS CLI.
2. Exécutez la commande `register-delegated-administrator`.

```
$ aws organizations register-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Exécutez la commande `list-delegated-administrators` pour vérifier que le compte membre spécifié en tant qu'administrateur délégué est correctement enregistré.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Pour annuler l'enregistrement d'un administrateur délégué (AWS CLI)

1. Ouvrez le AWS CLI.
2. Exécutez la commande `deregister-delegated-administrator`.

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Exécutez la commande `list-delegated-administrators` pour vérifier que l'enregistrement du compte membre spécifié en tant qu'administrateur délégué est correctement annulé.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Vous pouvez enregistrer à nouveau ce compte à tout moment.

Commencez à StackSets utiliser un exemple de modèle

Ce didacticiel vous aidera à commencer à StackSets utiliser le AWS Management Console. Il vous guide dans la création d'un à StackSet l'aide d'un exemple de modèle. Vous apprendrez à déployer des stacks dans plusieurs régions, à surveiller les StackSet opérations et à consulter les résultats.

Dans ce didacticiel, vous allez créer un StackSet qui active AWS Config Compte AWS dans les régions USA Ouest (Oregon) (us-west-2) et USA Est (Virginie du Nord) (us-east-1). Vous pouvez ainsi créer, mettre à jour ou supprimer des piles sur plusieurs comptes et régions en une seule opération, ce qui en fait une solution idéale pour gérer l'infrastructure à grande échelle. StackSets Bien que ce didacticiel utilise un seul compte pour des raisons de simplicité, il démontre efficacement les fonctionnalités multirégionales de StackSets.

L'exemple de modèle est disponible dans le compartiment S3 suivant : <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/Enable AWSConfig .yaml>.

 Note

StackSets est gratuit, mais les AWS ressources que vous créez avec ce logiciel vous seront facturées, comme AWS Config dans ce didacticiel. Pour en savoir plus, consultez [Pricing AWS Config](#) (Tarification).

Rubriques

- [Conditions préalables](#)
- [Créez un StackSet avec un exemple de modèle à partir de la console](#)
- [StackSet Création de moniteurs](#)
- [Afficher les StackSet résultats](#)
- [Mettez à jour votre StackSet](#)
- [Ajoutez des piles à votre StackSet](#)
- [Nettoyage](#)
- [Étapes suivantes](#)

Conditions préalables

Avant de commencer ce didacticiel, assurez-vous d'avoir rempli les conditions préalables suivantes :

- Vous devez avoir configuré les rôles IAM requis pour les autorisations autogérées. Pour créer StackSet et déployer des stacks au sein d'un seul compte, vous devez disposer des rôles suivants dans votre compte :
 - `AWSCloudFormationStackSetAdministrationRole`
 - `AWSCloudFormationStackSetExecutionRole`

Pour obtenir des instructions détaillées sur la configuration de ces rôles, consultez [Accorder des autorisations autogérées](#).

Créez un StackSet avec un exemple de modèle à partir de la console

Pour créer un StackSet qui permet AWS Config

1. Ouvrez la [CloudFormation console](#).

2. Dans la barre de navigation en haut de l'écran, choisissez Région AWS celui à StackSet partir duquel vous souhaitez gérer.

Vous pouvez choisir n'importe quelle région qui prend en charge StackSets. La région que vous sélectionnez n'a aucune incidence sur les régions dans lesquelles vous pouvez effectuer le déploiement avec votre StackSet.

3. Dans le volet de navigation, choisissez StackSets.
4. En haut de la StackSetspage, choisissez Create StackSet.
5. Sous Autorisations, sélectionnez Autorisations en libre-service, puis choisissez les rôles IAM que vous avez créés dans les prérequis.
 - Pour le rôle d'administrateur IAM, choisissez `AWSCloudFormationStackSetAdministrationRole`.
 - Pour le nom du rôle d'exécution IAM, sélectionnez `AWSCloudFormationStackSetExecutionRole`.
6. Sous Prerequisite - Prepare template (Prérequis - Préparer le modèle), choisissez Use a sample template (Utiliser un exemple de modèle).
7. Sous Sélectionner un modèle d'exemple, choisissez le modèle Activer AWS Config. Ensuite, choisissez Suivant.

Ce modèle crée les ressources nécessaires à activer AWS Config dans votre compte, notamment un enregistreur de configuration et un canal de diffusion.

8. Sur la page Spécifier les StackSet détails, saisissez le StackSet nom **my-awsconfig-stackset**.
9. Pour StackSet la description, entrez **A StackSet that enables Config across multiple Regions**.
10. Sous Paramètres, configurez les AWS Config paramètres comme suit :
 - a. Pour Support de tous les types de ressources, conservez la valeur par défaut, true, pour enregistrer tous les types de ressources pris en charge.
 - b. Pour Inclure les types de ressources globaux, conservez la valeur par défaut, false, pour exclure les ressources globales telles que les rôles IAM.
 - c. Laissez la liste des types de ressources si tous ne sont pas pris en charge définie sur <All>.
 - d. Pour La région contenant la ressource de rôle liée au service Config, remplacez < DeployToAnyRegion > par. **us-west-2**

Cela signifie que le rôle lié au service nommé `AWSServiceRoleForConfig` ne sera créé que si une pile est déployée dans la région USA Ouest (Oregon). Vous choisirez les régions de déploiement plus tôt lors de cette procédure.

- e. Pour la fréquence d'enregistrement de l'enregistreur de configuration, choisissez l'enregistrement QUOTIDIEN.

11. Choisissez Next (Suivant) pour continuer.

12. Sur la page des StackSet options de configuration, choisissez Ajouter une nouvelle balise et ajoutez une balise en spécifiant une paire clé/valeur :

- a. Pour Clé, entrez **Stage**.
- b. Pour le champ Value (Valeur), entrez **Test**.

Les balises auxquelles vous postulez StackSets sont appliquées aux ressources créées par vos piles.

13. Pour la configuration de l'exécution, choisissez Active pour activer CloudFormation la gestion optimisée des opérations :

- Les opérations non conflictuelles s'exécutent simultanément pour accélérer les temps de déploiement.
- Les opérations en conflit sont automatiquement mises en file d'attente et traitées dans l'ordre dans lequel elles ont été demandées.

Lorsque les opérations sont en cours ou en file d'attente, met en CloudFormation file d'attente toutes les opérations entrantes, même si elles ne sont pas conflictuelles. Vous ne pouvez pas modifier les paramètres d'exécution pendant cette période.

14. Choisissez Suivant.

15. Sur la page Définir les options de déploiement, pour Ajouter des piles à StackSet, choisissez Déployer de nouvelles piles.

16. Pour Accounts (Comptes), choisissez Deploy stacks in accounts (Déployer des piles dans des comptes).

17. Dans la zone de texte, entrez votre Compte AWS identifiant.

18. Pour Spécifier les régions, sélectionnez les régions suivantes dans cet ordre :

1. Région USA Ouest (Oregon) (`us-west-2`)

2. Région USA Est (Virginie du Nord) (us-east-1)

Utilisez la flèche vers le haut à côté de la région USA Ouest (Oregon) pour la déplacer en haut de la liste si nécessaire. L'ordre des régions détermine leur ordre de déploiement.

19. Pour les options de déploiement, configurez les paramètres suivants :

- a. Pour Maximum concurrent accounts (Nombre maximal de comptes en parallèle), conservez les valeurs par défaut Number (Nombre) et 1.

Pour les déploiements multicomptes, ce paramètre signifie que votre stack est CloudFormation déployé sur un seul compte à la fois.

- b. Pour Failure tolerance (Tolérance aux pannes), conservez les valeurs par défaut Number (Nombre) et 0.

Cela signifie qu'un maximum de déploiements zéro stack peuvent échouer dans l'une des régions que vous avez spécifiées avant d' CloudFormation arrêter le déploiement dans la région actuelle et d'annuler les déploiements dans les régions restantes.

- c. Pour la simultanéité des régions, choisissez Séquentiel (par défaut).

Ce paramètre garantit que les CloudFormation déploiements sont terminés dans une région avant de passer à la suivante.

- d. Pour le mode simultané, conservez la valeur par défaut de tolérance stricte aux défaillances.

Pour les déploiements multicomptes, cela réduit le niveau de simultanéité des comptes en cas de défaillance, tout en respectant la tolérance de défaillance +1.

20. Choisissez Suivant.

21. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.

22. Lorsque vous êtes prêt à créer votre StackSet, choisissez Soumettre.

StackSet Création de moniteurs

Une fois que vous avez choisi Soumettre CloudFormation , vous commencez à créer vos piles StackSet et à les déployer dans les régions spécifiées dans votre compte. La page de StackSet détails s'ouvre automatiquement et vous permet de suivre la progression de l'opération.

Pour suivre la StackSet création

1. Sur la page de StackSet détails, l'onglet Opérations est affiché par défaut, indiquant l'opération en cours.
2. Le statut de l'opération doit être RUNNING initialement. CloudFormation crée des piles dans les régions que vous avez spécifiées en fonction des options de déploiement que vous avez configurées.
3. Pour afficher plus de détails sur l'opération, sélectionnez l'ID d'opération dans la liste.
4. Sur la page Détails des opérations, vous pouvez afficher l'état des instances de piles créées dans chaque région.
5. Attendez que le statut de l'opération passe à SUCCEEDED, ce qui indique que toutes StackSet les instances de pile ont été créées avec succès.

Afficher les StackSet résultats

Une fois la StackSet création terminée, vous pouvez consulter les instances de stack déployées et vérifier qu' AWS Config elles ont été activées dans votre compte dans les régions spécifiées.

Pour consulter les StackSet résultats

1. Sur la page de StackSet détails, choisissez l'onglet Stack instances.
2. Vous devriez voir une liste des instances de stack créées dans votre compte dans les régions spécifiées. Chaque instance de pile doit avoir un état de SUCCEEDED, indiquant qu'elle a été déployée avec succès.
3. Pour vérifier que cela AWS Config est activé dans votre compte, vous pouvez consulter la AWS Config console dans chacune des régions déployées.

Mettez à jour votre StackSet

Après avoir créé votre StackSet, vous souhaitez peut-être le mettre à jour pour modifier les valeurs des paramètres ou ajouter d'autres régions. Cette section explique comment mettre à jour le paramètre de fréquence AWS Config d'enregistrement.

Pour mettre à jour votre StackSet

1. Sur la StackSet page, sélectionnez votre **my-awsconfig-stackset**.

2. Lorsque cette option est StackSet sélectionnée, choisissez Modifier StackSet les détails dans le menu Actions.
3. Sur la page Choisir un modèle, pour Condition préalable – Préparer le modèle, sélectionnez Utiliser le modèle actuel.
4. Choisissez Suivant.
5. Sur la page Spécifier StackSet les détails, sous Paramètres, recherchez la fréquence d'enregistrement de l'enregistreur de configuration et changez-la de QUOTIDIENNE à CONTINUE.
6. Choisissez Suivant.
7. Sur la page StackSet des options de configuration, laissez les paramètres tels quels et choisissez Suivant.
8. Sur la page Définir les options de déploiement, spécifiez votre identifiant de compte et les mêmes régions que celles que vous avez utilisées lors de la création du StackSet.
9. Pour les options de déploiement, conservez les mêmes paramètres qu'auparavant.
10. Choisissez Suivant.
11. Sur la page Vérification, vérifiez vos choix et choisissez Soumettre.
12. CloudFormation commence à mettre à jour votre StackSet. Vous pouvez suivre la progression dans l'onglet Opérations de la page de StackSet détails.

Ajoutez des piles à votre StackSet

Vous pouvez ajouter d'autres piles à votre compte StackSet en déployant dans d'autres régions. Cette section explique comment ajouter des piles dans une nouvelle région.

Pour ajouter des piles à votre StackSet

1. Sur la StackSetspage, sélectionnez votremy-awsconfig-stackset.
2. Lorsque cette option est StackSet sélectionnée, choisissez Ajouter des piles à dans le StackSet menu Actions.
3. Sur la page Définir les options de déploiement, pour Ajouter des piles à StackSet, choisissez Déployer de nouvelles piles.
4. Pour les comptes, choisissez Déployer des piles dans les comptes et entrez votre identifiant de compte.

5. Pour Spécifier les régions, sélectionnez une nouvelle région, telle que Europe (Irlande) (eu-west-1).
6. Pour les options de déploiement, conservez les mêmes paramètres qu'auparavant.
7. Choisissez Suivant.
8. Sur la page Spécifier les remplacements, laissez les valeurs des propriétés telles qu'elles sont spécifiées et sélectionnez Suivant.
9. Sur la page Vérification, vérifiez vos choix et choisissez Submit (Soumettre).
10. CloudFormation commence à créer de nouvelles piles dans la région spécifiée. Vous pouvez suivre la progression dans l'onglet Opérations de la page de StackSet détails.

Nettoyage

Pour éviter de payer des frais pour AWS Config des ressources indésirables, vous devez faire le ménage en supprimant les piles de votre fichier StackSet, en supprimant la pile StackSet elle-même et en supprimant les rôles IAM que vous avez créés pour ce didacticiel. Toutes les ressources étant déployées au sein de votre compte, le nettoyage est simple.

Pour supprimer des piles de votre StackSet

1. Sur la StackSetspage, sélectionnez votremy-awsconfig-stackset.
2. Lorsque cette option est StackSet sélectionnée, choisissez Supprimer les piles dans le StackSet menu Actions.
3. Sur la page Options de déploiement, sous Comptes, sélectionnez Déployer les piles dans les comptes et entrez votre ID de compte.
4. Pour Spécifier les régions, sélectionnez toutes les régions dans lesquelles vous avez déployé des piles.
5. Pour les options de déploiement, conservez les paramètres par défaut.
6. Assurez-vous que l'option Conserver les piles n'est pas activée, afin que les piles et leurs ressources soient supprimées.
7. Choisissez Suivant.
8. Sur la page Vérification, vérifiez vos choix et choisissez Submit (Soumettre).
9. CloudFormation commence à supprimer les piles de votre. StackSet Vous pouvez suivre la progression dans l'onglet Opérations de la page de StackSet détails.

Pour supprimer votre StackSet

1. Une fois que toutes les piles ont été supprimées, sur la StackSetspage, sélectionnez votremy-**awsconfig-stackset**.
2. Lorsque cette option est StackSet sélectionnée, choisissez Supprimer dans le StackSet menu Actions.
3. Lorsque vous êtes invité à confirmer, sélectionnez Supprimer.

Suppression des rôles du service IAM

Comme vous n'avez déployé que sur votre compte, vous n'avez qu'à supprimer les rôles IAM de ce seul compte, ce qui simplifie considérablement le nettoyage par rapport aux déploiements multi-comptes.

1. Ouvrez la [console IAM](#).
2. Dans le volet de navigation, choisissez Rôles.
3. Dans la zone de recherche, saisissez **AWSCloudFormationStackSet** pour trouver les rôles que vous avez créés pour ce didacticiel.
4. Cochez la case à côté de. AWSCloudFormationStackSetAdministrationRole
5. En haut de la page, choisissez Supprimer.
6. Dans le champ de confirmation, entrez **delete**, puis choisissez Supprimer.
7. Répétez le même processus pour supprimer le AWSCloudFormationStackSetExecutionRole.

Une fois le supprimé StackSet, un compartiment Amazon S3 restera dans chacun d'eux Région AWS en raison de l'DeletionPolicyattribut de la AWS::S3::Bucket ressource. Cela préserve vos données d' AWS Config historique. Si vous n'avez plus besoin de ces données, vous pouvez supprimer le compartiment manuellement en toute sécurité. Avant de pouvoir supprimer un compartiment, vous devez d'abord supprimer son contenu. Vider un compartiment supprime tous les objets qu'il contient.

Pour vider et supprimer les compartiments Amazon S3

1. Ouvrez la [console Amazon S3](#).
2. Dans le volet de navigation situé sur le côté gauche de la console, choisissez Compartiments.

3. Dans la liste des compartiments, vous verrez des compartiments créés à cet effet StackSet dans chaque région où vous avez effectué un déploiement. Sélectionnez l'option à côté du nom du bucket créé à cet effet StackSet, puis choisissez Empty.
4. Sur la page Vider le compartiment, confirmez que vous souhaitez vider le compartiment en saisissant **permanently delete** dans le champ de texte, puis choisissez Vider.
5. Surveillez la progression du processus de vidage du compartiment sur la page Vider le compartiment : état.
6. Pour revenir à votre liste de compartiments, choisissez Quitter.
7. Sélectionnez l'option en regard du nom du compartiment, puis choisissez Supprimer.
8. Lorsque vous êtes invité à confirmer, saisissez le nom du compartiment, puis sélectionnez Supprimer le compartiment.
9. Surveillez la progression du processus de suppression des compartiments à partir de la liste Compartiments. Lorsque Amazon S3 a terminé la suppression du compartiment, il le supprime de la liste.
10. Répétez ce processus pour chaque compartiment créé par les StackSet dans les différentes régions.

Étapes suivantes

Félicitations ! Vous avez réussi à en créer un StackSet à l'aide d'un exemple de modèle, à déployer des piles dans plusieurs régions de votre compte, à les mettre à jour StackSet, à en ajouter d'autres et à nettoyer vos ressources. En vous concentrant sur le déploiement d'un seul compte, vous avez simplifié le processus de nettoyage tout en apprenant les principales fonctionnalités multirégionales de StackSets

Pour en savoir plus StackSets, consultez les rubriques suivantes :

- [Remplacez les valeurs des paramètres sur les piles de votre CloudFormation StackSet](#) : découvrez comment remplacer les valeurs des paramètres pour des comptes et des régions spécifiques.
- [Créez CloudFormation StackSets avec des autorisations gérées par le service](#)— Découvrez la création StackSets pour les déploiements multi-comptes avec AWS Organizations

Créez CloudFormation StackSets avec des autorisations autogérées

Cette rubrique explique comment créer à l'aide de StackSets des autorisations autogérées pour déployer des piles dans différentes régions AWS et régions.

Note

Avant de continuer, créez les rôles de service IAM requis StackSets pour établir une relation de confiance entre le compte à StackSet partir duquel vous administrez et le compte sur lequel vous déployez des stacks. Pour de plus amples informations, veuillez consulter [Accorder des autorisations autogérées](#).

Rubriques

- [Créer un StackSet avec des autorisations autogérées \(console\)](#)
- [Créez un StackSet avec des autorisations autogérées \(AWS CLI\)](#)

Créer un StackSet avec des autorisations autogérées (console)

Pour créer un StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez Région AWS celui à StackSet partir duquel vous souhaitez gérer.
3. Dans le volet de navigation, choisissez StackSets.
4. En haut de la StackSetspage, choisissez Create StackSet.
5. Sous Autorisations, choisissez Autorisations en libre-service et choisissez les rôles IAM que vous avez créés.
6. Sous Prérequis – Préparer le modèle, sélectionnez Le modèle est prêt.
7. Sous Spécifier le modèle, choisissez soit de spécifier l'URL du compartiment S3 qui contient votre modèle de pile, soit de charger un fichier de modèle de pile. Ensuite, choisissez Suivant.
8. Sur la page Spécifier StackSet les détails, attribuez un nom au StackSet, spécifiez les paramètres éventuels, puis choisissez Next.

9. Choisissez Next (Suivant) pour continuer.
10. Sur la page des StackSet options de configuration, sous Balises, spécifiez les balises à appliquer aux ressources de votre pile. Pour plus d'informations sur la façon dont les balises sont utilisées AWS, consultez la section [Organisation et suivi des coûts à l'aide des balises de répartition des AWS coûts](#) dans le Guide de AWS Billing and Cost Management l'utilisateur.
11. Pour la configuration de l'exécution, choisissez Active pour activer CloudFormation la gestion optimisée des opérations :
 - Les opérations non conflictuelles s'exécutent simultanément pour accélérer les temps de déploiement.
 - Les opérations en conflit sont automatiquement mises en file d'attente et traitées dans l'ordre dans lequel elles ont été demandées.

Lorsque les opérations sont en cours ou en file d'attente, met en CloudFormation file d'attente toutes les opérations entrantes, même si elles ne sont pas conflictuelles. Vous ne pouvez pas modifier les paramètres d'exécution pendant cette période.

12. Si votre modèle contient des ressources IAM, dans la section Capacités, choisissez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous autorisez l'utilisation de ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
13. Choisissez Suivant.
14. Sur la page Définir les options de déploiement, pour Ajouter des piles à StackSet, choisissez Déployer de nouvelles piles.
15. Pour Accounts (Comptes), choisissez Deploy stacks in accounts (Déployer des piles dans des comptes). Collez vos Compte AWS numéros cibles dans la zone de texte, en séparant les numéros multiples par des virgules.

 Note

Vous pouvez inclure votre identifiant de compte administrateur si vous souhaitez également déployer des stacks dans ce compte.

16. Sous Spécifier les régions, choisissez les régions dans lesquelles vous voulez déployer des piles.
17. Pour Options de déploiement, procédez comme suit :

- Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
- Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
- Pour la simultanéité des régions, choisissez comment traiter les régions : séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).
- Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.
 - Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.

18. Choisissez Suivant.

19. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.

20. Lorsque vous êtes prêt à créer votre StackSet, choisissez Soumettre.

CloudFormation commence à créer votre StackSet. Consultez la progression et le statut de la création des piles StackSet dans votre page de StackSet détails qui s'ouvre lorsque vous choisissez Soumettre.

Créez un StackSet avec des autorisations autogérées ()AWS CLI

Suivez les étapes décrites dans cette section pour utiliser le AWS CLI pour :

- Créez le StackSet conteneur.
- Déployez des instances de pile.

Pour créer un StackSet

1. Utilisez la [create-stack-set](#) commande pour créer un nouveau StackSet nom *my-stackset*. L'exemple suivant utilise un modèle stocké dans un compartiment S3 et inclut un paramètre qui définit un *KeyPairName* avec la valeur *TestKey*.

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/  
MyApp.template \  
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Une fois votre `create-stack-set` commande terminée, exécutez-la pour vérifier que votre [list-stack-sets](#) commande StackSet a été créée. Vous devriez voir votre nouveau StackSet dans les résultats.

```
aws cloudformation list-stack-sets
```

3. Utilisez la [create-stack-instances](#) commande pour déployer des piles dans votre StackSet. L'exemple suivant déploie des piles dans deux Comptes AWS (*account_ID_1* et *account_ID_2*) dans deux régions (*us-west-2* et *us-east-1*).

Définissez le traitement simultané des comptes et les autres préférences de déploiement à l'aide de cette option `--operation-preferences`. Cet exemple utilise des paramètres basés sur le décompte. Notez que `MaxConcurrentCount` ne doit pas dépasser `FailureToleranceCount + 1`. Pour les paramètres basés sur des pourcentages, utilisez `FailureTolerancePercentage` ou `MaxConcurrentPercentage` à la place.

```
aws cloudformation create-stack-instances \  
  --stack-set-name my-stackset \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour plus d'informations, consultez [CreateStackInstances](#) dans la Référence d'API AWS CloudFormation .

4. Utilisez la commande [describe-stack-set-operation](#) pour vérifier que vos piles ont été créées avec succès. Pour l'option `--operation-id`, spécifiez l'ID d'opération renvoyé dans le cadre de la sortie `create-stack-instances`.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Créez CloudFormation StackSets avec des autorisations gérées par le service

Avec les autorisations gérées par les services, vous pouvez déployer des piles sur des comptes gérés par AWS Organizations des régions spécifiques. Avec ce modèle, il n'est pas nécessaire de créer des rôles IAM dans chaque compte cible et Région AWS. CloudFormation crée les rôles IAM en votre nom. Pour de plus amples informations, veuillez consulter [Activer l'accès approuvé](#).

Rubriques

- [Considérations](#)
- [Créer un StackSet avec des autorisations gérées par le service \(console\)](#)
- [Créer un StackSet avec des autorisations gérées par le service \(AWS CLI\)](#)

Considérations

Avant de créer un StackSet avec des autorisations gérées par le service, tenez compte des points suivants :

- StackSets avec des autorisations gérées par les services peuvent être initiées soit par le compte de gestion de votre organisation, soit par des comptes d'administrateur délégué, mais toutes les opérations sont effectuées par le compte de gestion.
- CloudFormation ne déploie pas les piles sur le compte de gestion, même si ce compte fait partie de votre organisation ou appartient à une unité organisationnelle (UO).
- StackSet Vous pouvez cibler l'ensemble de votre organisation (y compris tous les comptes) ou certains d'entre eux OUs. Lorsqu'une unité d'organisation StackSet cible un parent, elle inclut automatiquement n'importe quel enfant OUs. Par défaut, lorsqu'une StackSet cible est spécifique OUs, elle inclut tous les comptes concernés OUs. Cependant, vous pouvez cibler des comptes spécifiques à l'aide des options de filtrage des comptes.
- Plusieurs StackSets peuvent cibler la même organisation ou la même unité d'organisation.
- Vous ne pouvez pas cibler des comptes en dehors de votre organisation.
- Votre autorisation de déploiement StackSets dépend des autorisations attribuées au principal IAM (utilisateur, rôle ou groupe) que vous utilisez pour vous connecter au compte de gestion. Pour obtenir un exemple de politique IAM qui accorde des autorisations de déploiement dans une organisation, veuillez consulter [Restriction des opérations sur les ensembles de piles en fonction de la région et des types de ressources](#).

- Les administrateurs délégués disposent de toutes les autorisations nécessaires pour effectuer des déploiements sur n'importe quel compte de votre organisation. Le compte de gestion ne peut pas limiter les autorisations d'administrateur déléguées pour le déploiement sur des StackSet opérations OUs ou des opérations spécifiques.
- Les paramètres de déploiement automatique s'appliquent au StackSet niveau. Vous ne pouvez pas ajuster les déploiements automatiques de manière sélective pour OUs, les comptes ou les régions.
- StackSets qui utilisent des autorisations gérées par les services ne prennent pas en charge les piles imbriquées ou les modèles contenant des macros ou des transformations.

Créer un StackSet avec des autorisations gérées par le service (console)

Pour créer un StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez Région AWS celui à StackSet partir duquel vous souhaitez gérer.
3. Dans le volet de navigation, choisissez StackSets.
4. En haut de la StackSetspage, choisissez Create StackSet.
5. Sous Autorisations, choisissez Autorisations gérées par le service.

Note

Si l'accès sécurisé avec AWS Organizations est désactivé, une bannière s'affiche. Un accès sécurisé est requis pour créer ou mettre à jour un StackSet avec des autorisations gérées par le service. Seul l'administrateur du compte de gestion de l'organisation dispose des autorisations nécessaires pour [Activez un accès sécurisé pour StackSets avec AWS Organizations](#).

6. Sous Prérequis – Préparer le modèle, sélectionnez Le modèle est prêt.
7. Sous Spécifier le modèle, choisissez soit de spécifier l'URL du compartiment S3 qui contient votre modèle de pile, soit de charger un fichier de modèle de pile. Ensuite, choisissez Suivant.
8. Sur la page Spécifier StackSet les détails, attribuez un nom au StackSet, spécifiez les paramètres éventuels, puis choisissez Next.

9. Sur la page des StackSet options de configuration, sous Balises, spécifiez les balises à appliquer aux ressources de votre pile. Pour plus d'informations sur la façon dont les balises sont utilisées AWS, consultez la section [Organisation et suivi des coûts à l'aide des balises de répartition des AWS coûts](#) dans le Guide de AWS Billing and Cost Management l'utilisateur.
10. Pour la configuration de l'exécution, choisissez Active pour activer CloudFormation la gestion optimisée des opérations :
 - Les opérations non conflictuelles s'exécutent simultanément pour accélérer les temps de déploiement.
 - Les opérations en conflit sont automatiquement mises en file d'attente et traitées dans l'ordre dans lequel elles ont été demandées.

Lorsque les opérations sont en cours ou en file d'attente, met en CloudFormation file d'attente toutes les opérations entrantes, même si elles ne sont pas conflictuelles. Vous ne pouvez pas modifier les paramètres d'exécution pendant cette période.

11. Si votre modèle contient des ressources IAM, dans la section Capacités, choisissez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous autorisez l'utilisation de ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
12. Choisissez Suivant pour continuer et activer l'accès sécurisé si cela n'a pas encore été fait.
13. Sur la page Définir les options de déploiement, sous Cibles de déploiement, procédez de l'une des manières suivantes :
 - Pour déployer vers tous les comptes de votre organisation, choisissez Déployer vers l'organisation.
 - Pour effectuer un déploiement sur tous les comptes en particulier OUs, choisissez Déployer vers les unités organisationnelles (OUs). Choisissez Ajouter une UO, puis collez l'ID d'unité d'organisation cible dans la zone de texte. Répétez l'opération pour chaque nouvelle UO cible.

Si vous avez choisi Déployer vers les unités organisationnelles (OUs), pour le type de filtre de compte, vous pouvez définir vos cibles de déploiement pour qu'elles soient des comptes individuels spécifiques en choisissant l'une des options suivantes et en fournissant des numéros de compte.

- Aucun (par défaut) : déployez des piles sur tous les comptes de la zone spécifiée OUs.

- Intersection — Déployez des piles sur des comptes individuels spécifiques au sein des comptes sélectionnés OUs.
 - Différence — Déployez des piles sur tous les comptes sélectionnés, à l'exception de comptes spécifiques.
 - Union — Déployez des piles sur les comptes spécifiés OUs ainsi que sur des comptes individuels supplémentaires.
14. Sous Déploiement automatique, choisissez si vous souhaitez effectuer le déploiement automatique sur les comptes ajoutés à l'organisation cible ou OUs dans le futur. Pour de plus amples informations, veuillez consulter [Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations](#).
15. Si vous avez activé le déploiement automatique, sous Comportement de suppression de compte, choisissez si les ressources de pile sont conservées ou supprimées lorsqu'un compte est supprimé d'une organisation cible ou d'une OU.

 Note

Lorsque l'option Conserver les piles est sélectionnée, les piles sont supprimées de vos piles StackSet, mais les piles et les ressources associées sont conservées. Les ressources restent dans leur état actuel, mais n'en feront plus partie StackSet.

16. Sous Spécifier les régions, choisissez les régions dans lesquelles vous voulez déployer des piles.
17. Pour Options de déploiement, procédez comme suit :
- Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
 - Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
 - Pour la simultanéité des régions, choisissez comment traiter les régions : séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).
 - Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.

- Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.
- Pour StackSet les dépendances, ajoutez dependent StackSet ARNs, en respectant un maximum de 10 dépendances. Pour de plus amples informations, veuillez consulter [Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations](#).

18. Choisissez Next (Suivant) pour continuer.

19. Sur la page de révision, vérifiez que vous StackSet allez effectuer le déploiement sur les bons comptes dans les bonnes régions, puis choisissez Créer StackSet.

La page StackSet de détails s'ouvre. Vous pouvez consulter la progression et le statut de la création des piles dans votre StackSet.

Créer un StackSet avec des autorisations gérées par le service ()AWS CLI

Suivez les étapes décrites dans cette section pour utiliser le AWS CLI pour :

- Créez le StackSet conteneur.
- Déployez des instances de pile.

Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

Deploy to your organization

Pour créer un StackSet

1. Utilisez la [create-stack-set](#) commande pour créer un nouveau StackSet nom *my-stackset*. L'exemple suivant utilise un modèle stocké dans un compartiment S3, active les déploiements automatiques et conserve les piles lorsque les comptes sont supprimés. Pour de plus amples informations, veuillez consulter [Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations](#).

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url s3://my-bucket/my-template.yaml
```

```
--template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/  
MyApp.template \  
--permission-model SERVICE_MANAGED \  
--auto-deployment  
Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

2. Utilisez la [list-stack-sets](#) commande pour confirmer que vous avez bien StackSet été créé. Votre nouveau StackSet est répertorié dans les résultats.

```
aws cloudformation list-stack-sets
```

- Si vous définissez l'--call-asoption sur DELEGATED_ADMIN lorsque vous êtes connecté à votre compte de membre, list-stack-sets tous les fichiers dotés d'autorisations gérées StackSets par le service sont renvoyés dans le compte de gestion de l'organisation.
 - Si vous définissez l'--call-asoption sur SELF lorsque vous êtes connecté à votre Compte AWS, list-stack-sets renvoie tous les éléments autogérés StackSets dans votre Compte AWS.
 - Si vous définissez l'--call-asoption sur SELF lorsque vous êtes connecté au compte de gestion de l'organisation, toutes les données sont list-stack-sets renvoyées StackSets dans le compte de gestion de l'organisation.
3. Utilisez la [create-stack-instances](#) commande pour ajouter des piles à votre StackSet. Pour l'option --deployment-targets, spécifiez l'ID racine de l'organisation afin de déployer sur tous les comptes de votre organisation.

Définissez le traitement simultané des comptes et les autres préférences de déploiement à l'aide de cette option --operation-preferences. Cet exemple utilise des paramètres basés sur le décompte. Notez que MaxConcurrentCount ne doit pas dépasser FailureToleranceCount + 1. Pour les paramètres basés sur des pourcentages, utilisez FailureTolerancePercentage ou MaxConcurrentPercentage à la place.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=r-a1b2c3d4e5 \  
--regions us-west-2 us-east-1 \  
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour plus d'informations, consultez [CreateStackInstances](#) dans la Référence d'API AWS CloudFormation .

4. À l'aide du `operation-id` renvoyé dans la sortie `create-stack-instances`, utilisez la commande [describe-stack-set-operation](#) suivante pour vérifier que vos piles ont été créées avec succès.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to organizational units (OUs)

Pour créer un StackSet

1. Utilisez la [create-stack-set](#) commande pour créer un nouveau StackSet nom *my-stackset*. L'exemple suivant utilise un modèle stocké dans un compartiment S3 et inclut un paramètre qui définit un *KeyPairName* avec la valeur *TestKey*

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Utilisez la [list-stack-sets](#) commande pour confirmer que vous avez bien StackSet été créé. Votre nouveau StackSet est répertorié dans les résultats.

```
aws cloudformation list-stack-sets
```

- Si vous définissez l'option `--call-as` sur `DELEGATED_ADMIN` lorsque vous êtes connecté à votre compte de membre, `list-stack-sets` tous les fichiers dotés d'autorisations gérées StackSets par le service sont renvoyés dans le compte de gestion de l'organisation.
- Si vous définissez l'option `--call-as` sur `SELF` lorsque vous êtes connecté à votre Compte AWS, `list-stack-sets` renvoie tous les éléments autogérés StackSets dans votre Compte AWS.
- Si vous définissez l'option `--call-as` sur `SELF` lorsque vous êtes connecté au compte de gestion de l'organisation, toutes les données sont `list-stack-sets` renvoyées StackSets dans le compte de gestion de l'organisation.

3. Utilisez la [create-stack-instances](#) commande pour ajouter des piles à votre StackSet. Pour l'option `--deployment-targets`, spécifiez l'unité d'organisation IDs vers laquelle le déploiement doit être effectué.

Définissez le traitement simultané des comptes et les autres préférences de déploiement à l'aide de cette option `--operation-preferences`. Cet exemple utilise des paramètres basés sur le décompte. Notez que `MaxConcurrentCount` ne doit pas dépasser `FailureToleranceCount + 1`. Pour les paramètres basés sur des pourcentages, utilisez `FailureTolerancePercentage` ou `MaxConcurrentPercentage` à la place.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slrl5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour plus d'informations, consultez [CreateStackInstances](#) dans la Référence d'API AWS CloudFormation .

4. À l'aide du `operation-id` renvoyé dans la sortie `create-stack-instances`, utilisez la commande [describe-stack-set-operation](#) suivante pour vérifier que vos piles ont été créées avec succès.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

Vous pouvez cibler des unités organisationnelles spécifiques (OUs) et utiliser le filtrage des comptes pour contrôler précisément quels comptes reçoivent des déploiements de stack. Par défaut, les piles sont déployées sur tous les comptes compris dans les limites spécifiées OUs si aucun filtrage de compte n'est spécifié.

Dans le AWS CLI, vous spécifiez le filtrage des comptes avec l'option `--deployment-targets`. Pour de plus amples informations, veuillez consulter [DeploymentTargets](#).

Après avoir créé le StackSet conteneur à l'aide de la `create-stack-set` commande, utilisez l'un des exemples suivants pour déployer des piles sur des comptes spécifiques.

Cibler des comptes spécifiques dans une UO

L'exemple suivant déploie des piles uniquement sur les comptes A1 et A2 de OU1

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets  
OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=INTERSECTION \  
--regions us-west-2 us-east-1
```

Exclure des comptes d'une UO

L'exemple suivant déploie des piles sur tous les comptes OU1 sauf les comptes A1 et A2.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets  
OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=DIFFERENCE \  
--regions us-west-2 us-east-1
```

Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations

CloudFormation peut déployer automatiquement des piles supplémentaires sur de nouveaux AWS Organizations comptes lorsqu'ils sont ajoutés à votre organisation ou à vos unités organisationnelles cibles (OUs). Vous pouvez activer les déploiements automatiques et choisir de supprimer ou de conserver les piles et les ressources associées lorsque des comptes sont supprimés de Target. OUs Ces paramètres peuvent être modifiés à tout moment pour utiliser StackSets les autorisations gérées par le service.

Fonctionnement des déploiements automatiques

Lorsque les déploiements automatiques sont activés, ils sont déclenchés lorsque des comptes sont ajoutés à une organisation ou à une unité d'organisation cible, supprimés d'une organisation ou d'une unité d'organisation cible, ou déplacés d'une cible OUs à une autre.

Prenons l'exemple de StackSet1 qui cible OU1 dans la région us-east-1 et de StackSet2 qui cible OU2 dans la région us-east-1. OU1 contient AccountA.

Si vous passez AccountA de OU1 à OU2 avec les déploiements automatiques activés, exécute CloudFormation automatiquement une opération de suppression pour retirer la StackSet1 pile AccountA et met en file d'attente une opération de création qui ajoute la StackSet2 pile à AccountA

Considérations

Voici quelques considérations à prendre en compte lors de l'utilisation des déploiements automatiques :

- La fonctionnalité de déploiements automatiques est activée au StackSet niveau. Vous ne pouvez pas ajuster les déploiements automatiques de manière sélective pour OUs, les comptes ou les régions.
- Les valeurs de paramètres remplacées ne s'appliquent qu'aux comptes qui se trouvent actuellement dans la cible OUs et à leur enfant. OUs Les comptes ajoutés à la cible OUs et à son enfant OUs à l'avenir utiliseront les valeurs StackSet par défaut et non les valeurs remplacées.
- Les déploiements automatiques ne prennent pas en compte les filtres de ciblage au niveau du compte. Si vous ciblez des comptes spécifiques et activez les déploiements automatiques, vous StackSet continuerez à déployer sur tous les comptes nouvellement ajoutés au sein de l'organisation déployée. Pour empêcher les déploiements vers les comptes nouvellement ajoutés, désactivez les déploiements automatiques.
- Gestion des dépendances : définissez jusqu'à 10 dépendances par compte StackSet, avec un maximum de 100 dépendances par compte. Par exemple, si vous en avez cinq StackSets avec cinq dépendances chacune, vous avez 25 dépendances prises en compte dans la limite de 100 dépendances. Vous pouvez demander une augmentation de limite via la [console de quotas de service](#). Les dépendances sont supprimées lorsqu'elles StackSets sont supprimées ou lorsque les Organisations sont désactivées.
- Il est recommandé d'activer l'exécution gérée lors de l'utilisation de déploiements automatiques. L'exécution gérée permettra aux opérations de déploiement automatique sur plusieurs comptes cibles de s'exécuter simultanément au sein d'un même compte StackSet, augmentant ainsi la vitesse de traitement, en particulier pour les grandes Organisations.

Activer ou désactiver les déploiements automatiques (console)

Pour activer ou désactiver les déploiements automatiques

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Dans le volet de navigation, choisissez StackSets.

4. Sur la StackSetspage, sélectionnez l'option à côté du nom de la personne StackSet à mettre à jour.
5. Sélectionnez Modifier le déploiement automatique dans le menu Actions situé dans le coin supérieur droit.
6. Dans la boîte de dialogue qui s'ouvre, procédez comme suit :
 - a. Pour Déploiement automatique, sélectionnez Activé ou Désactivé.
 - b. Pour Comportement de suppression de compte, sélectionnez Supprimer les piles ou Conserver les piles. Les ressources conservées restent dans leur état actuel, mais n'en feront plus partie StackSet.
 - c. Pour StackSet les dépendances, ajoutez dependent StackSet ARNs, en respectant un maximum de 10 dépendances.
7. Choisissez Enregistrer.

Activer ou désactiver les déploiements automatiques (AWS CLI)

Pour activer ou désactiver les déploiements automatiques

1. Utilisez la commande [update-stack-set](#) avec l'option `--auto-deployment`.

La commande suivante active les déploiements automatiques.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

Pour désactiver les déploiements automatiques, spécifiez `Enabled=false` comme valeur pour l'option `--auto-deployment`, comme dans l'exemple suivant.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment Enabled=false
```

2. À l'aide de l'ID d'opération renvoyé dans le cadre de la `update-stack-set` sortie, exécutez [describe-stack-set-operation](#) pour vérifier que votre mise à jour StackSet a été effectuée avec succès.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```


Mettre à jour CloudFormation StackSets

Vous pouvez effectuer la mise à jour à l' StackSet aide de la CloudFormation console ou du AWS CLI.

Pour ajouter et supprimer des comptes et des régions dans un StackSet, consultez [Ajoutez des piles à StackSets](#) et [Supprimer des piles de StackSets](#). Pour remplacer les valeurs des paramètres d'une pile, consultez [Remplacer les paramètres sur les piles](#).

Rubriques

- [Mettez à jour votre StackSet \(console\)](#)
- [Mettez à jour votre StackSet \(AWS CLI\)](#)

Mettez à jour votre StackSet (console)

Pour mettre à jour un StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Dans le volet de navigation, choisissez StackSets.
4. Sur la StackSetspage, sélectionnez celui que StackSet vous souhaitez mettre à jour.
5. Lorsque cette option est StackSet sélectionnée, choisissez Modifier StackSet les détails dans le menu Actions.
6. Sur la page Choisissez un modèle, mettez à jour la section Autorisations si nécessaire, ou passez à l'étape suivante.
7. Pour Prérequis – Préparer le modèle, choisissez Utiliser le modèle actuel pour utiliser le modèle actuel, ou Remplacer le modèle actuel pour spécifier une URL S3 vers un autre modèle ou charger un nouveau modèle.
8. Choisissez Suivant.
9. Sur la page Spécifier StackSet les détails, pour la StackSet description, mettez à jour la description StackSet selon les besoins.
10. Pour Paramètres, mettez à jour les valeurs des paramètres selon vos besoins.
11. Choisissez Suivant.

12. Sur la page des StackSet options de configuration, pour les balises, modifiez les balises selon vos besoins. Vous pouvez ajouter, mettre à jour ou supprimer des balises. Pour plus d'informations sur la façon dont les balises sont utilisées AWS, consultez la section [Organisation et suivi des coûts à l'aide des balises de répartition des AWS coûts](#) dans le Guide de AWS Billing and Cost Management l'utilisateur.
13. Pour Configuration d'exécution, vous pouvez mettre à jour la configuration d'exécution selon vos besoins.

 Note

N'oubliez pas que vous ne pouvez pas modifier les paramètres d'exécution lorsque des opérations sont en cours d'exécution ou en attente.

14. Si votre modèle contient des ressources IAM, dans la section Capacités, choisissez Je sais que ce modèle peut créer des ressources IAM pour indiquer que vous autorisez l'utilisation de ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).
15. Choisissez Suivant.
16. Sur la page Définir les options de déploiement, indiquez les comptes et les régions pour la mise à jour.

CloudFormation déploiera les mises à jour de la pile dans les comptes spécifiés au sein de la première région, puis passera à la suivante, et ainsi de suite, tant que les échecs de déploiement d'une région ne dépassent pas la tolérance d'échec spécifiée.

- a. [Autorisations autogérées] Pour Comptes, Emplacements de déploiement, choisissez Déployer des piles dans les comptes. Collez le compte cible IDs que vous avez utilisé pour créer le votre StackSet dans la zone de texte, en séparant les nombres par des virgules.

[Autorisations gérées par le service] Effectuez l'une des méthodes suivantes :

- Choisissez Déployer vers les unités organisationnelles (OUs). Entrez la cible OUs que vous avez utilisée pour créer votre StackSet.
 - Choisissez Déployer sur les comptes. Collez l'unité d'organisation IDs ou le compte cible IDs que vous avez utilisé pour créer votre StackSet.
- b. Pour Spécifier les régions, spécifiez l'ordre dans lequel vous souhaitez CloudFormation déployer vos mises à jour.

- c. Pour Options de déploiement, procédez comme suit :
- Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
 - Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
 - Pour la simultanéité des régions, choisissez comment traiter les régions : séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).
 - Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.
 - Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.
 - [Autorisations gérées par le service] Pour les StackSet dépendances, ajoutez une dépendance StackSet ARNs, en respectant un maximum de 10 dépendances. Pour de plus amples informations, veuillez consulter [Activer ou désactiver les déploiements automatiques pour StackSets dans AWS Organizations](#).
- d. Choisissez Next (Suivant) pour continuer.
17. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.
18. Lorsque vous êtes prêt à continuer, sélectionnez Soumettre.

CloudFormation commence à appliquer vos mises à jour à votre StackSet et affiche l'onglet Opérations de la page de StackSet détails. Vous pouvez afficher la progression et le statut des opérations de mise à jour dans l'onglet Operations.

Mettez à jour votre StackSet (AWS CLI)

Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

1. Pour mettre à jour un StackSet

Utilisez la [update-stack-set](#) commande pour apporter des modifications à votre StackSet.

Dans les exemples suivants, nous mettons à jour l'--parameters option StackSet en utilisant. Plus précisément, nous modifions la fréquence de livraison par défaut des instantanés pour la configuration du canal de livraison de TwentyFour_Hours à Twelve_Hours. Comme nous utilisons toujours le modèle actuel, nous ajoutons l'option --use-previous-template.

Définissez le traitement simultané des comptes et les autres préférences de déploiement à l'aide de cette option --operation-preferences. Ces exemples utilisent des paramètres basés sur le décompte. Notez que MaxConcurrentCount ne doit pas dépasser FailureToleranceCount + 1. Pour les paramètres basés sur des pourcentages, utilisez FailureTolerancePercentage ou MaxConcurrentPercentage à la place.

[Autorisations autogérées] Pour --accounts cette option, indiquez le compte que IDs vous souhaitez cibler par votre mise à jour.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

[Autorisations gérées par le service] Pour --deployment-targets cette option, indiquez l'ID racine de l'organisation ou l'unité organisationnelle (UO) que IDs vous souhaitez cibler par votre mise à jour.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour plus d'informations, consultez [UpdateStackSet](#) dans la Référence d'API AWS CloudFormation .

2. Vérifiez que vous avez bien StackSet été mis à jour en exécutant la `describe-stack-set-operation` commande pour afficher l'état et les résultats de votre opération de mise à jour. Pour `--operation-id`, utilisez l'ID d'opération qui a été renvoyé par la commande `update-stack-set`.

```
aws cloudformation describe-stack-set-operation \  
--operation-id operation_ID
```

Ajoutez des piles à CloudFormation StackSets

Lorsque vous créez un StackSet, vous pouvez créer les piles correspondantes. StackSet CloudFormation vous permet également d'ajouter d'autres piles, pour des comptes et des régions supplémentaires, à tout moment après leur StackSet création. Vous pouvez ajouter des piles à l'aide de la CloudFormation console ou du AWS CLI.

Rubriques

- [Ajouter des piles à une StackSet \(console\)](#)
- [Ajouter des piles à un StackSet \(\)AWS CLI](#)

Ajouter des piles à une StackSet (console)

Pour ajouter des piles à un StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Dans le volet de navigation, choisissez StackSets. Sur la StackSets page, sélectionnez celui StackSet que vous avez créé.
4. Lorsque cette option est StackSet sélectionnée, choisissez Ajouter des piles à dans le StackSet menu Actions.
5. Sur la page Définir les options de déploiement, procédez comme suit :
 - a. Pour Ajouter des piles à StackSet, choisissez Déployer de nouvelles piles.
 - b. Procédez ensuite comme suit en fonction de la configuration StackSet de vos autorisations :

- [Autorisations autogérées] Pour Comptes, Emplacements de déploiement, choisissez Déployer des piles dans les comptes. Collez vos numéros de compte de destination dans la zone de texte, en séparant les numéros par des virgules.
- [Autorisations gérées par le service] Pour Cibles de déploiement, procédez de l'une des manières suivantes :
 - Choisissez Déployer dans l'organisation pour déployer sur tous les comptes de votre organisation.
 - Choisissez Déployer vers les unités organisationnelles (OUs) pour effectuer le déploiement sur tous les comptes en particulier OUs. Choisissez Ajouter une autre UO, puis collez l'ID d'unité d'organisation cible dans la zone de texte. Répétez l'opération pour chaque nouvelle unité organisationnelle cible. CloudFormation cible également n'importe quel enfant parmi OUs les cibles que vous avez sélectionnées.

 Note

Si vous ajoutez une unité d'organisation que vous StackSet ciblez déjà, vous CloudFormation créez de nouvelles piles dans tous les comptes de l'unité d'organisation qui ne possèdent pas encore de piles StackSet (par exemple, les comptes qui ont été ajoutés à l'unité d'organisation après la création de la vôtre et dont les déploiements automatiques sont désactivés). StackSet

- c. Pour Spécifier les régions, indiquez les Régions AWS dans lesquelles déployer les comptes cibles que vous avez spécifiés à l'étape précédente. Par défaut, déploie CloudFormation les piles dans les comptes spécifiés au sein de la première région, puis passe à la suivante, et ainsi de suite, tant que les échecs de déploiement d'une région ne dépassent pas la tolérance d'échec spécifiée.
- d. Pour Options de déploiement, procédez comme suit :
 - Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
 - Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
 - Pour la simultanéité des régions, choisissez comment traiter les régions : séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).

- Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.
 - Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.
 - e. Choisissez Suivant.
6. Sur la page Specify Overrides (Spécifier des remplacements), laissez les valeurs de propriétés telles qu'indiquées. Vous ne remplacerez pas les valeurs de propriétés pour les piles que vous allez créer. Choisissez Suivant.
 7. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.
 8. Lorsque vous êtes prêt à continuer, sélectionnez Soumettre.

CloudFormation commence à créer vos piles. Consultez la progression et le statut de la création des piles StackSet dans votre page de StackSet détails qui s'ouvre lorsque vous choisissez Soumettre. Une fois terminé, vos nouvelles piles doivent être répertoriées dans l'onglet Instances de pile.

Ajouter des piles à un StackSet ()AWS CLI

Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

Pour ajouter des piles à un StackSet avec des autorisations autogérées

Utilisez la commande `create-stack-instances` de l'interface de ligne de commande. Pour l'option `--accounts`, indiquez les comptes IDs pour lesquels vous souhaitez créer des piles.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--accounts account_id --regions eu-west-1 us-west-2
```

Pour ajouter des piles à un StackSet avec des autorisations gérées par le service

Utilisez la commande `create-stack-instances` de l'interface de ligne de commande. Pour `--deployment-targets` cette option, indiquez l'ID de l'organisation (racine) ou l'unité d'organisation IDs pour laquelle vous souhaitez créer des piles. Pour des exemples de commandes ciblant des comptes spécifiques, consultez [Créer un StackSet avec des autorisations gérées par le service \(AWS CLI\)](#).

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-  
west-2
```

Note

Si vous ajoutez une unité d'organisation que vous StackSet ciblez déjà, vous CloudFormation créez de nouvelles piles dans tous les comptes de l'unité d'organisation qui ne possèdent pas encore de piles StackSet (par exemple, les comptes qui ont été ajoutés à l'unité d'organisation après la création de la vôtre et dont les déploiements automatiques sont désactivés). StackSet

Remplacez les valeurs des paramètres sur les piles de votre CloudFormation StackSet

Dans certains cas, vous souhaitez peut-être que les piles de certaines régions ou de certains comptes aient des valeurs de propriété différentes de celles spécifiées dans le fichier StackSet lui-même. Par exemple, vous pourriez spécifier une autre valeur pour un paramètre donné selon l'objectif de développement ou de production d'un compte. Dans ces situations, vous CloudFormation permet de remplacer les valeurs des paramètres dans les piles par compte et par région. Vous pouvez remplacer les valeurs des paramètres du modèle lors de la création initiale des piles, et vous pouvez remplacer les valeurs des paramètres pour les piles existantes. Vous pouvez uniquement rétablir les paramètres que vous avez précédemment remplacés dans les piles aux valeurs spécifiées dans le StackSet.

Les remplacements de valeurs de paramètres s'appliquent aux piles dans les comptes et les régions que vous sélectionnez. Pendant les StackSet mises à jour, les valeurs de paramètres remplacées pour une pile ne sont pas mises à jour, mais conservent leur valeur remplacée.

Vous ne pouvez remplacer que les valeurs des paramètres spécifiées dans le StackSet ; pour ajouter ou supprimer un paramètre lui-même, vous devez mettre à jour le StackSet modèle. Si vous ajoutez

un paramètre à un StackSet modèle, avant de pouvoir remplacer cette valeur de paramètre dans une pile, vous devez d'abord mettre à jour toutes les piles avec le nouveau paramètre et la nouvelle valeur spécifiés dans le. StackSet Une fois que toutes les piles ont été mises à jour avec le nouveau paramètre, vous pouvez alors remplacer la valeur du paramètre dans les piles individuelles comme vous le souhaitez.

Pour savoir comment remplacer les valeurs des StackSet paramètres lorsque vous créez des piles, consultez. [Ajoutez des piles à StackSets](#)

Rubriques

- [Remplacer les paramètres sur les piles \(console\)](#)
- [Remplacer les paramètres sur les piles \(AWS CLI\)](#)

Remplacer les paramètres sur les piles (console)

Pour remplacer les paramètres de piles spécifiques

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Dans le volet de navigation, choisissez StackSets. Sur la StackSets page, sélectionnez votre StackSet.
4. Lorsque cette StackSet option est sélectionnée, choisissez Remplacer StackSet les paramètres dans le menu Actions.
5. Sur la page Options de déploiement, indiquez les comptes et les régions pour les piles pour lesquelles vous allez créer des remplacements.

Par défaut, déploie CloudFormation les piles dans les comptes spécifiés au sein de la première région, puis passe à la suivante, et ainsi de suite, à condition que les échecs de déploiement d'une région ne dépassent pas la tolérance d'échec spécifiée.

- a. [Autorisations autogérées] Pour les Emplacements de déploiement, choisissez Déployer les piles dans les comptes. Collez une partie ou la totalité du compte cible IDs que vous avez utilisé pour créer votre StackSet.

[Autorisations gérées par le service] Effectuez l'une des méthodes suivantes :

- Choisissez Déployer vers les unités organisationnelles (OUs). Entrez une ou plusieurs cibles OUs que vous avez utilisées pour créer votre StackSet. Les valeurs des paramètres remplacées ne s'appliquent qu'aux comptes qui se trouvent actuellement dans la cible OUs et à leur enfant. OUs Les comptes ajoutés à la cible OUs et à son enfant OUs à l'avenir utiliseront les valeurs StackSet par défaut et non les valeurs remplacées.
 - Choisissez Déployer sur les comptes. Collez une partie ou la totalité de l'unité d' IDs organisation ou du compte cible IDs que vous avez utilisé pour créer votre StackSet.
- b. Pour Spécifier les régions, ajoutez une ou plusieurs des régions dans lesquelles vous avez déployé des piles à cet StackSet effet.

Si vous ajoutez plusieurs régions, l'ordre des régions sous Specify regions (Spécifier des régions) détermine l'ordre de leur déploiement.

- c. Pour Options de déploiement, procédez comme suit :
- Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
 - Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
 - Pour la simultanéité des régions, choisissez comment traiter les régions :
 - séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).
 - Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.
 - Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.
- d. Choisissez Suivant.
6. Sur la page Spécifier les remplacements, cochez les cases correspondant aux paramètres à remplacer, puis choisissez Remplacer la valeur dans le menu Modifier la StackSet valeur de remplacement.
7. Sur la page Remplacer les valeurs des StackSet paramètres, apportez vos modifications, puis choisissez Enregistrer les modifications.

 Note

Pour rétablir les paramètres remplacés en utilisant la valeur spécifiée dans le StackSet, vérifiez tous les paramètres et choisissez Définir sur la valeur dans le menu Modifier la StackSet valeur de remplacement. Cela supprime toutes les valeurs remplacées une fois que vous avez mis à jour les piles.

8. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.
9. Lorsque vous êtes prêt à continuer, sélectionnez Soumettre.

CloudFormation commence à mettre à jour vos piles. Consultez la progression et le statut des piles sur la page de StackSet détails qui s'ouvre lorsque vous choisissez Soumettre.

Remplacer les paramètres sur les piles (AWS CLI)

 Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

Pour remplacer les paramètres de piles spécifiques

1. Utilisez la [update-stack-instances](#) AWS CLI commande et spécifiez l'`--parameter-overrides` option.

[Autorisations autogérées] Pour `--accounts` cette option, indiquez le compte IDs pour lequel vous souhaitez remplacer les valeurs des paramètres sur les piles.

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --accounts account_id --regions us-east-1
```

[Autorisations gérées par le service] Pour `--deployment-targets` cette option, indiquez l'ID racine de l'organisation, l'unité d'organisation ou le AWS Organizations compte IDs pour

lequel vous souhaitez remplacer les paramètres sur les piles. IDs Dans cet exemple, nous remplaçons les valeurs des paramètres pour les piles dans tous les comptes de l'UO avec l'ID *ou-rcuk-1x5j1lwo*.

Les valeurs des paramètres remplacées s'appliquent uniquement aux comptes qui se trouvent actuellement dans l'unité d'organisation cible et à son enfant. OUs Les comptes ajoutés à l'unité d'organisation cible et OUs à son enfant à l'avenir utiliseront les valeurs StackSet par défaut et non les valeurs remplacées.

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo \  
  --regions us-east-1
```

2. Vérifiez que vos valeurs de paramètres ont été remplacées avec succès sur les piles en exécutant la commande `describe-stack-set-operation` pour afficher l'état et les résultats de votre opération de mise à jour. Pour `--operation-id`, utilisez l'ID d'opération qui a été renvoyé par la commande `update-stack-instances`.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Supprimer des piles de CloudFormation StackSets

Vous pouvez supprimer des piles à StackSets l'aide de la CloudFormation console ou du AWS CLI.

Rubriques

- [Supprimer des piles de votre StackSet \(console\)](#)
- [Supprimer les piles de votre StackSet \(AWS CLI\)](#)

Note

La suppression de piles dans une unité organisationnelle (UO) de niveau supérieur entraîne la suppression de cette UO en tant StackSet que cible.

Supprimer des piles de votre StackSet (console)

Pour supprimer les piles

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Dans le volet de navigation, choisissez StackSets. Sur la StackSets page, sélectionnez le StackSet.
4. Lorsque vous StackSet êtes sélectionné, choisissez Supprimer les piles dans le StackSet menu Actions.
5. Sur la page Options de déploiement du set, sélectionnez d'abord les comptes et les régions dans lesquels vous souhaitez supprimer les piles.
 - a. [Autorisations autogérées] Pour Comptes, choisissez Déployer des piles dans les comptes ou Déployer des piles dans les unités organisationnelles.

Si vous choisissez Déployer les piles dans les comptes, collez vos numéros de compte cibles dans la zone de texte Numéros de compte, en séparant les numéros multiples par des virgules.

Si vous choisissez Déployer les piles dans les unités organisationnelles, collez un ID d'unité organisationnelle cible dans la zone de texte Numéros d'organisation pour cibler tous les comptes qui font partie de l'organisation spécifiée.

- b. [Autorisations gérées par le service] Pour les unités organisationnelles (OUs), spécifiez l'unité d'organisation cible. IDs

Important

CloudFormation supprimera les piles de la cible spécifiée OUs et de son enfant OUs.

Pour le type de filtre Compte, vous pouvez affiner les comptes dont les piles seront supprimées en choisissant l'une des options suivantes et en fournissant les numéros de compte.

- Aucun (par défaut) — Supprime les piles de tous les comptes spécifiés OUs.
 - Intersection — Supprime les piles uniquement à partir de comptes individuels spécifiques au sein des comptes sélectionnés OUs.
 - Différence — Supprime les piles de tous les comptes sélectionnés, à l'exception de comptes spécifiques.
 - Union — Supprime les piles des comptes individuels spécifiés OUs ainsi que des comptes individuels supplémentaires.
- c. Pour Spécifier les régions, sélectionnez les régions dans lesquelles vous souhaitez supprimer des piles au sein des comptes cibles.
6. Pour Options de déploiement, procédez comme suit :
- Pour Nombre maximal de comptes simultanés, spécifiez le nombre de comptes traités simultanément.
 - Pour Tolérance aux défaillances, spécifiez le nombre maximum de défaillances de compte autorisées par région. L'opération s'arrêtera et ne se poursuivra pas dans d'autres régions une fois cette limite atteinte.
 - Pour conserver les piles, activez cette option pour enregistrer les piles et leurs ressources associées lorsque vous les supprimez de votre StackSet. Les ressources restent dans leur état actuel mais n'en font plus partie StackSet.
 - Pour la simultanéité des régions, choisissez comment traiter les régions : séquentiellement (une région à la fois) ou en parallèle (plusieurs régions simultanément).
 - Pour le mode simultané, choisissez le comportement de la simultanéité lors de l'exécution de l'opération.
 - Tolérance stricte aux pannes : réduit le niveau de concurrence des comptes en cas de panne, en restant dans les limites de la tolérance aux pannes +1.
 - Tolérance aux pannes légères : maintient le niveau de concurrence spécifié (la valeur du nombre maximal de comptes simultanés) indépendamment des pannes.
7. Choisissez Suivant.
8. Sur la page Vérification, vérifiez vos choix. Pour apporter des modifications, sélectionnez Modifier dans la section correspondante.
9. Lorsque vous êtes prêt à retirer les piles de votre ordinateur StackSet, choisissez Soumettre.

Une fois la suppression des piles terminée, vous pouvez vérifier que les piles ont été supprimées de votre compte StackSet dans la page StackSet détaillée, dans l'onglet Stack instances.

Supprimer les piles de votre StackSet ()AWS CLI

Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

Utilisez la `delete-stack-instances` commande avec votre StackSet nom.

Dans ces exemples, nous utilisons l'option `--no-retain-stacks` car nous ne conservons pas de piles. Utilisez `--retain-stacks` à la place de `--no-retain-stacks` si vous souhaitez conserver les piles et leurs ressources.

Pour `--regions`, spécifiez le dont Régions AWS vous souhaitez supprimer les piles, par exemple, `us-west-2` et `us-east-1`.

Définissez le traitement simultané des comptes et les autres préférences à l'aide de cette option `--operation-preferences`. Ces exemples utilisent des paramètres basés sur le décompte. Notez que `MaxConcurrentCount` ne doit pas dépasser `FailureToleranceCount` + 1. Pour les paramètres basés sur des pourcentages, utilisez `FailureTolerancePercentage` ou `MaxConcurrentPercentage` à la place.

Pour supprimer des piles (autorisations autogérées)

Pour `--accounts` cette option, spécifiez IDs le compte dont vous souhaitez supprimer les piles.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --no-retain-stacks \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour supprimer des piles (autorisations gérées par le service)

Pour `--deployment-targets`, spécifiez l'ID racine de l'organisation ou l'unité organisationnelle (UO) IDs dont vous souhaitez supprimer les piles.

 Important

CloudFormation supprimera les piles de la cible spécifiée OUs et de son enfant OUs.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=ou-rcuk-1x5jlwo,ou-rcuk-slr5lh0a \  
--regions us-west-2 us-east-1 \  
--no-retain-stacks \  
--operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Pour plus d'informations, consultez [DeleteStackInstances](#) dans la Référence d'API AWS CloudFormation .

Facultativement, une fois la suppression des piles terminée, vérifiez que les piles ont été supprimées de votre StackSet compte en exécutant la `describe-stack-set-operation` commande pour afficher le statut et les résultats de l'opération de suppression des piles. Pour `--operation-id`, utilisez l'ID d'opération qui a été renvoyé par la commande `delete-stack-instances`.

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \  
--operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```

Supprimer CloudFormation StackSets

Pour supprimer un StackSet, vous devez d'abord supprimer toutes les piles du StackSet. Pour plus d'informations sur la manière de supprimer toutes les piles, consultez [Supprimer des piles de StackSets](#).

Rubriques

- [Supprimer une StackSet \(console\)](#)
- [Supprimer un StackSet \(AWS CLI\)](#)
- [Supprimer les rôles de service \(facultatif\)](#)

Supprimer une StackSet (console)

Pour supprimer un StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le StackSet dans lequel Région AWS vous l'avez créé.
3. Sur la StackSetspage, sélectionnez le StackSet.
4. Lorsque cette option est StackSet sélectionnée, choisissez Supprimer dans le StackSet menu Actions.
5. Lorsque vous êtes invité à confirmer que vous souhaitez supprimer le StackSet, choisissez Supprimer StackSet.

Supprimer un StackSet (AWS CLI)

Note

Lorsque vous agissez en tant qu'administrateur délégué, vous devez inclure `--call-as DELEGATED_ADMIN` dans la commande.

Pour supprimer un StackSet

1. Utilisez la commande `delete-stack-set` suivante. Lorsque vous êtes invité à confirmer, tapez **y**, puis appuyez sur Entrée.

```
aws cloudformation delete-stack-set --stack-set-name my-stackset
```

2. Vérifiez que le StackSet a été supprimé en exécutant la `list-stack-sets` commande. Les résultats de la `list-stack-sets` commande devraient afficher votre pile avec un statut de `DELETED`.

```
aws cloudformation list-stack-sets
```

Supprimer les rôles de service (facultatif)

Si vous n'avez plus besoin des rôles de service IAM nécessaires pour effectuer StackSet des opérations, nous vous recommandons de supprimer les rôles. CloudFormation

Pour l'autogestion StackSets, les rôles que vous avez créés. Pour plus d'informations sur ces rôles, consultez [Accorder des autorisations autogérées](#).

Pour les rôles gérés par des services StackSets, les rôles créés automatiquement StackSets ont le suffixe CloudFormationStackSetsOrgAdmin dans le compte de gestion de l'organisation et CloudFormationStackSetsOrgMember dans chaque compte cible. Pour de plus amples informations, veuillez consulter [Rôles liés à un service](#).

Pour supprimer un rôle de service (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le panneau de navigation, choisissez Rôles, puis cochez la case en regard du rôle à supprimer.
3. Dans le menu Actions du rôle en haut de la page, choisissez Supprimer le rôle.
4. Dans la boîte de dialogue de confirmation, choisissez Yes, Delete (Oui, supprimer). Si vous êtes sûr, procédez à la suppression même si les dernières données consultées dans le service sont toujours en cours de chargement.

Pour supprimer un rôle de service (AWS CLI)

- Utilisez la commande delete-role suivante. Lorsque vous êtes invité à confirmer, tapez **y**, puis appuyez sur Enter.

```
aws iam delete-role --role-name role name
```

Pour plus d'informations sur la suppression des rôles, consultez la section [Supprimer des rôles ou des profils d'instance](#) dans le Guide de l'utilisateur IAM.

Empêcher les StackSets déploiements échoués à l'aide des portes de compte cible

Une porte de compte est une fonctionnalité optionnelle qui vous permet de vérifier qu'un compte cible répond à certaines exigences avant de CloudFormation commencer StackSets les opérations sur ce compte. Cette vérification est effectuée par le biais d'une AWS Lambda fonction qui fait office de vérification préalable.

Un exemple courant de portail de compte consiste à vérifier qu'aucune CloudWatch alarme n'est active ou non résolue sur le compte cible. CloudFormation invoque la fonction Lambda chaque fois que vous lancez des opérations de stack dans le compte cible, et ne continue que si la fonction renvoie SUCCEEDED un code. Si la fonction Lambda renvoie un statut de FAILED, CloudFormation elle ne poursuit pas l'opération demandée. Si aucune fonction Lambda de blocage de compte n'est configurée CloudFormation , ignore la vérification et poursuit l'opération.

Si votre compte de destination échoue à une vérification du portail de compte, l'opération qui a échoué est prise en compte dans le nombre ou le pourcentage de piles spécifié dans la tolérance aux pannes. Pour plus d'informations sur la tolérance aux pannes, consultez [StackSet options de fonctionnement](#).

Le blocage des comptes n'est disponible que pour les StackSets opérations. Cette fonctionnalité n'est pas disponible pour d'autres CloudFormation opérations en dehors de StackSets.

Exigences

Les exigences suivantes doivent être satisfaites pour la fermeture d'un compte :

- Votre fonction Lambda doit être nommée `AWSCloudFormationStackSetAccountGate` pour utiliser cette fonctionnalité.
- Ils `AWSCloudFormationStackSetExecutionRole` doivent avoir les autorisations nécessaires pour appeler votre fonction Lambda. Sans ces autorisations, il CloudFormation ignorera la vérification du compte et procédera aux opérations de stack.
- Pour que la consignation de compte fonctionne, l'autorisation `Lambda InvokeFunction` doit être ajoutée aux comptes de destination. La politique d'approbation du compte de destination doit avoir une relation d'approbation avec le compte d'administrateur. Voici un exemple de déclaration de politique qui accorde des autorisations `Lambda InvokeFunction`.

JSON

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

CloudFormation modèles pour créer des fonctions Lambda

Utilisez les exemples de modèles suivants pour créer des fonctions

AWSCloudFormationStackSetAccountGate Lambda. Pour créer une pile à l'aide de l'un de ces modèles, consultez [Création d'une pile depuis la CloudFormation console](#).

Emplacement du modèle	Description
https://s3.amazonaws.com/cloudformation-stacks-sample-templates-us-east-1/AccountGateSucceeded.yml	Crée une pile qui implémente une fonction de consignment de compte Lambda qui renvoie le statut SUCCEEDED .
https://s3.amazonaws.com/cloudformation-stacks-sample-templates-us-east-1/AccountGateFailed.yml	Crée une pile qui implémente une fonction de consignment de compte Lambda qui renvoie le statut FAILED.

Choisissez le mode de simultanéité pour CloudFormation StackSets

Le mode de simultanéité est un paramètre [StackSetOperationPreferences](#) qui vous permet de choisir le comportement du niveau de simultanéité pendant les opérations. StackSet Vous pouvez choisir parmi les modes suivants :

- **Tolérance stricte aux pannes** : cette option réduit dynamiquement le niveau de simultanéité afin de garantir que le nombre de comptes défaillants ne dépasse jamais la valeur de Tolérance aux pannes + 1. La simultanéité réelle initiale est définie sur la valeur la plus basse entre la valeur du Nombre maximal de comptes en parallèle, ou la valeur de Tolérance aux pannes + 1. La simultanéité réelle est ensuite réduite proportionnellement au nombre de défaillances. Il s'agit du comportement de par défaut.
- **Tolérance souple aux pannes** : cette option dissocie la Tolérance aux pannes de la simultanéité réelle. Cela permet aux StackSet opérations de s'exécuter au niveau de simultanéité défini par la valeur Maximum de comptes simultanés, quel que soit le nombre d'échecs.

La tolérance stricte aux défaillances réduit la vitesse de déploiement en cas StackSet d'échec des opérations, car la simultanéité diminue à chaque échec. Soft Failure Tolerance donne la priorité à la vitesse de déploiement tout en tirant parti des capacités CloudFormation de sécurité. Cela vous permet d'examiner et de résoudre les défaillances StackSet opérationnelles pour les problèmes courants tels que ceux liés aux ressources existantes, aux quotas de service et aux autorisations.

Pour plus d'informations sur les échecs des opérations de StackSets pile, consultez [Raisons les plus courantes de l'échec de l'opération de pile](#).

Pour plus d'informations sur le Nombre maximal de comptes en parallèle et la Tolérance aux pannes, consultez [StackSet options de fonctionnement](#).

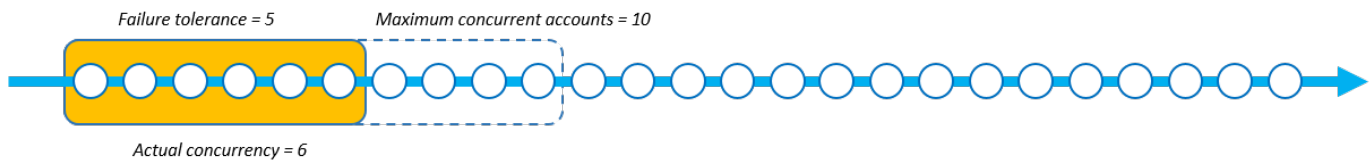
Fonctionnement de chaque mode simultané

Les images ci-dessous fournissent une représentation visuelle du fonctionnement de chaque mode de simultanéité au cours d'une StackSet opération. La chaîne de nœuds représente un déploiement vers un seul Région AWS et chaque nœud est une cible Compte AWS.

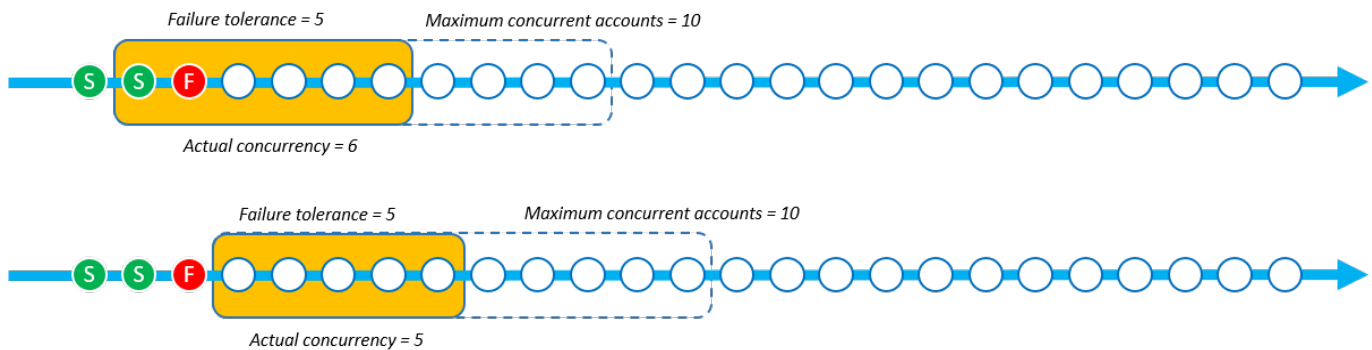
Tolérance stricte aux pannes

Lorsqu'une StackSet opération utilisant la tolérance d'échec stricte a une valeur de tolérance d'échec définie sur 5 et que la valeur maximale de comptes simultanés est définie sur 10, la simultanéité réelle est de 6. La simultanéité réelle est de 6, car cette valeur de Tolérance aux pannes de 5 + 1 est inférieure à la valeur du Nombre maximal de comptes en parallèle.

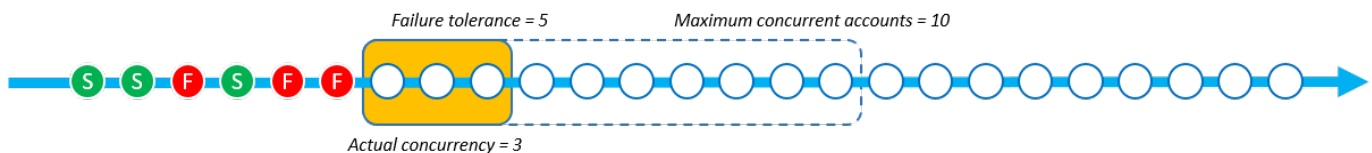
L'image suivante montre l'impact de la valeur de tolérance aux défaillances sur la valeur maximale des comptes simultanés, ainsi que l'impact de ces deux valeurs sur la simultanéité réelle de l' StackSet opération :



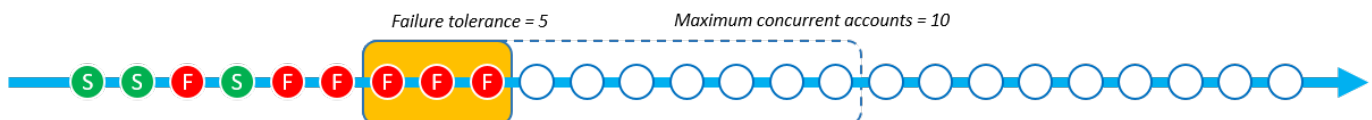
Lorsque le déploiement commence et que des instances de pile sont défaillantes, la simultanéité réelle est réduite pour garantir une expérience de déploiement sûre. La simultanéité réelle passe de 6 à 5 en cas d' StackSets échec du déploiement d'une instance de pile.



Le mode Tolérance stricte aux pannes réduit la simultanéité réelle proportionnellement au nombre d'instances de pile défaillantes. Dans l'exemple suivant, la simultanéité réelle passe de 5 à 3 en cas d' StackSets échec du déploiement de 2 instances de pile supplémentaires, portant le total des instances de pile défaillantes à 3.



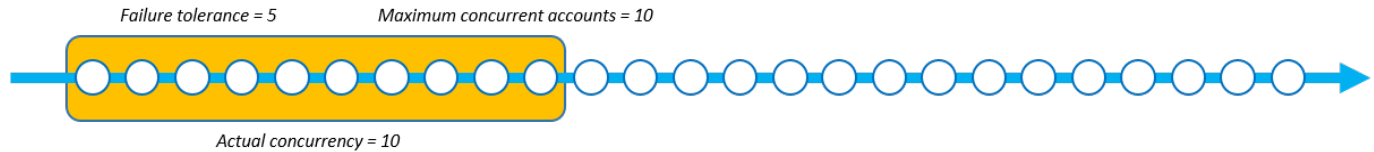
StackSets échoue lorsque le StackSet nombre d'instances de pile défaillantes est égal à la valeur définie de Tolérance de défaillance +1. Dans l'exemple suivant, l'opération StackSets échoue lorsqu'il y a 6 instances de pile défaillantes et que la valeur de tolérance de défaillance est de 5.



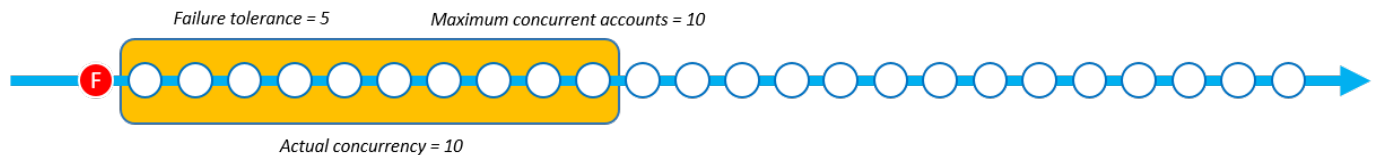
Dans cet exemple, 9 instances de stack CloudFormation ont été déployées (3 ont réussi et 6 ont échoué) avant d'arrêter l' StackSet opération.

Tolérance souple aux pannes

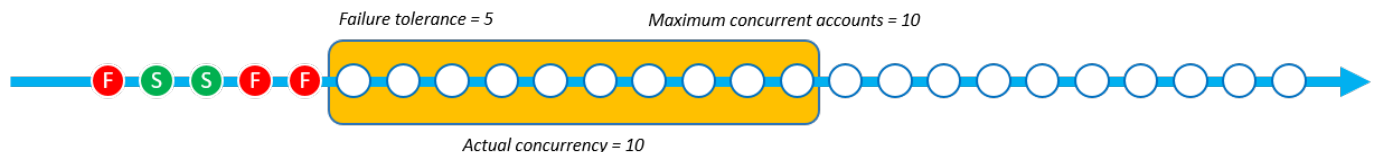
Lorsque la valeur de tolérance d'échec d'une StackSet opération utilisant Soft Failure Tolerance est définie sur 5 et que la valeur maximale de comptes simultanés est définie sur 10, la simultanéité réelle est de 10.



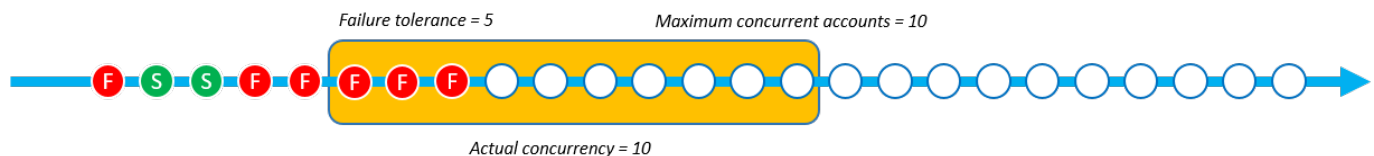
Lorsque le déploiement commence et que des instances de pile sont défaillantes, la simultanéité réelle ne change pas. Dans l'exemple suivant, une opération de pile est défaillante, mais la simultanéité réelle reste de 10.



La simultanéité réelle reste de 10 même après deux défaillances d'instances de pile supplémentaires.

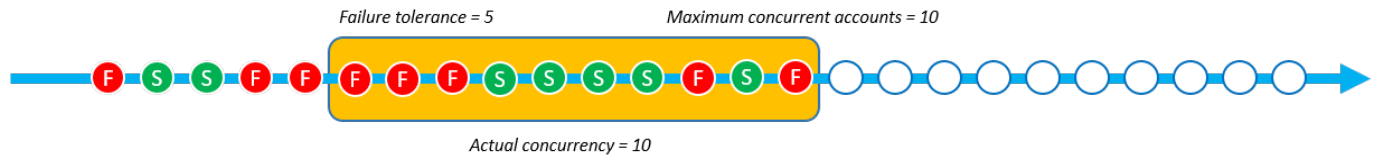


StackSets échoue lorsque les StackSet instances de pile défaillantes dépassent la valeur de tolérance de défaillance. Dans l'exemple suivant, l'opération StackSets échoue lorsqu'il y a 6 instances de pile défaillantes et que le nombre de tolérances d'échec est de 5. Toutefois, l'opération ne se terminera pas tant que les autres opérations de la file d'attente de simultanéité ne seront pas terminées.



StackSets continue de déployer des instances de pile qui se trouvent déjà dans la file d'attente de simultanéité. Cela signifie que le nombre d'instances de piles défaillantes peut être supérieur à la Tolérance aux pannes. Dans l'exemple suivant, 8 instances de pile ont échoué car la file d'attente

de simultanéité avait encore 7 opérations à effectuer, même si l' StackSet opération avait atteint la tolérance d'échec de 5.



Dans cet exemple, 15 instances de pile StackSets ont été déployées (7 ont réussi et 8 ont échoué) avant d'arrêter l'opération de pile.

Sélection de la tolérance stricte aux pannes et de la tolérance souple aux pannes en fonction de la vitesse de déploiement

Le choix entre les modes de tolérance de défaillance stricte et de tolérance de défaillance souple dépend de la vitesse préférée de votre StackSet déploiement et du nombre autorisé d'échecs de déploiement.

Les tableaux suivants montrent comment chaque mode de simultanéité gère une StackSet opération qui échoue lors d'une tentative de déploiement de 1 000 instances de pile au total. Dans chaque scénario, la valeur de tolérance aux pannes est fixée à 100 instances de pile et la valeur du Nombre maximal de comptes en parallèle est fixée à 250 instances de pile.

Bien que les files d'attente soient StackSets en fait des comptes sous forme de fenêtre coulissante (voir [Fonctionnement de chaque mode simultané](#)), cet exemple montre le fonctionnement par lots pour démontrer la rapidité de chaque mode.

Tolérance stricte aux pannes

Cet exemple d'utilisation du mode de Tolérance stricte aux pannes réduit la simultanéité réelle par rapport au nombre de défaillances survenues dans chaque lot précédent. Chaque lot comporte 20 instances défaillantes, ce qui réduit la simultanéité réelle du lot suivant de 20 jusqu'à ce que l' StackSet opération atteigne la valeur de tolérance d'échec de 100.

Dans le tableau suivant, la simultanéité initiale réelle du premier lot est de 101 instances de pile. La simultanéité réelle est de 101, car il s'agit de la valeur inférieure du Nombre maximal de comptes en parallèle (250) ou de la Tolérance aux pannes (100) + 1. Chaque lot contient 20 déploiements d'instances de pile défaillants, ce qui réduit ensuite la simultanéité réelle de chaque lot suivant de 20 instances de pile.

Tolérance stricte aux pannes	Lot 1	Lot 2	Lot 3	Lot 4	Lot 5	Lot 6
Nombre réel de simultanés	101	81	61	41	21	-
Nombre d'instances défaillantes	20	20	20	20	20	-
Nombre d'instances de piles réussies	81	61	41	21	1	-

L'opération utilisant la tolérance de défaillance stricte a permis d'effectuer 305 déploiements d'instances de pile en 5 lots au moment où l' StackSet opération a atteint la tolérance d'échec de 100 instances de pile. L' StackSet opération déploie avec succès 205 instances de pile avant d'échouer.

Tolérance souple aux pannes

Cet exemple d'utilisation du mode de Tolérance souple aux pannes maintient le même nombre de simultanés réelles définies par la valeur Nombre maximal de comptes en parallèle de 250 instances de pile, quel que soit le nombre d'instances défaillantes. Les StackSet opérations conservent la même simultanété réelle jusqu'à ce qu'elle atteigne la valeur de tolérance de défaillance de 100 instances.

Dans le tableau suivant, la simultanété initiale réelle du premier lot est de 250 instances de pile. La simultanété réelle est de 250 car la valeur maximale des comptes simultanés est définie sur 250 et le mode de tolérance de défaillance souple StackSets permet d'utiliser cette valeur comme simultanété réelle, quel que soit le nombre de défaillances. Même s'il y a 50 défaillances dans chacun des lots pour cet exemple, la simultanété réelle demeure inchangée.

Tolérance souple aux pannes	Lot 1	Lot 2	Lot 3	Lot 4	Lot 5	Lot 6
Nombre réel de simultanés	250	250	-	-	-	-
Nombre d'instances défaillantes	50	50	-	-	-	-
Nombre d'instances de piles réussies	200	200	-	-	-	-

En utilisant la même valeur Nombre maximal de comptes en parallèle et la même valeur de Tolérance aux pannes, l'opération utilisant le mode de Tolérance souple aux pannes a permis de réaliser 500 déploiements d'instances de pile en 2 lots. L' StackSet opération déploie avec succès 400 instances de pile avant d'échouer.

Choix du mode simultané (console)

Lors de la création ou de la mise à jour d'un StackSet, sur la page Définir les options de déploiement, pour le mode simultané, choisissez Tolérance de défaillance stricte ou Tolérance de défaillance souple.

Choix du mode simultané (AWS CLI)

Vous pouvez utiliser le `ConcurrencyMode` paramètre avec les StackSets commandes suivantes :

- [create-stack-instances](#)
- [delete-stack-instances](#)
- [detect-stack-set-drift](#)
- [import-stacks-to-stack-set](#)

- [update-stack-instances](#)
- [update-stack-set](#)

Ces commandes contiennent un paramètre existant appelé `--operation-preferences` qui peut utiliser le paramètre `ConcurrencyMode`. `ConcurrencyMode` peut être défini sur l'une des valeurs suivantes :

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

L'exemple suivant crée une instance de pile utilisant le `ConcurrencyMode STRICT_FAILURE_TOLERANCE`, avec un `FailureToleranceCount` défini sur 10 et un `MaxConcurrentCount` défini sur 5.

```
aws cloudformation create-stack-instances \  
  --stack-set-name example-stackset \  
  --accounts 123456789012 \  
  --regions eu-west-1 \  
  --operation-preferences  
  ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

Note

Pour connaître les procédures détaillées relatives à la création et à la mise à jour d'un `StackSet`, consultez les rubriques suivantes :

- [Créer StackSets \(autorisations autogérées\)](#)
- [Créer StackSets \(autorisations gérées par le service\)](#)
- [Mettre à jour StackSets](#)

Exécution d'une détection de dérive sur CloudFormation StackSets

Même si vous gérez vos piles et les ressources qu'elles contiennent CloudFormation, les utilisateurs peuvent modifier ces ressources en dehors de CloudFormation. Les utilisateurs peuvent modifier une ressource directement en utilisant le service sous-jacent qui a créé la ressource. En effectuant

une détection de dérive sur un StackSet, vous pouvez déterminer si l'une des instances de la pile appartenant à celui-ci StackSet diffère ou a dérivé de la configuration attendue.

Rubriques

- [Comment CloudFormation effectue la détection de la dérive sur un StackSet](#)
- [Détection de la dérive sur une StackSet \(console\)](#)
- [Détection de la dérive sur un StackSet \(AWS CLI\)](#)
- [Arrêter la détection de dérive sur un StackSet](#)

Comment CloudFormation effectue la détection de la dérive sur un StackSet

Lorsqu'il CloudFormation effectue une détection de dérive sur un StackSet, il effectue une détection de dérive sur la pile associée à chaque instance de pile du StackSet. Pour ce faire, CloudFormation compare l'état actuel de chaque ressource de la pile avec l'état attendu de cette ressource, tel que défini dans le modèle de la pile et tous les paramètres d'entrée spécifiés. Si l'état actuel d'une ressource diffère de l'état attendu, cette ressource est considérée comme présentant un écart. Si une ou plusieurs ressources d'une pile présentent un écart, alors la pile elle-même est considérée comme présentant un écart, et les instances de pile auxquelles la pile est associée sont également considérées comme présentant un écart. Si une ou plusieurs instances d'une pile StackSet ont dérivé, l'instance StackSet elle-même est considérée comme ayant dérivé.

La détection des dérives identifie les modifications non gérées, c'est-à-dire les modifications apportées aux piles situées en dehors de. CloudFormation Les modifications apportées directement CloudFormation à une pile, plutôt qu'au StackSet niveau, ne sont pas considérées comme de la dérive. Supposons, par exemple, que vous ayez une pile associée à une instance de pile de StackSet a. Si vous avez l'habitude de mettre à jour cette pile pour utiliser un modèle différent, cela n'est pas considéré comme une dérive, même si cette pile possède désormais un modèle différent de celui de toutes les autres piles appartenant au StackSet. En effet, la pile correspond toujours à la configuration de modèle et de paramètre attendue dans CloudFormation.

Pour des informations détaillées sur la manière dont CloudFormation la détection de la dérive sur une pile est effectuée, voir [Détection des modifications de configuration non gérées apportées aux piles et aux ressources grâce à la détection des écarts](#).

Comme il CloudFormation effectue une détection de dérive sur chaque pile individuellement, il prend en compte toutes les valeurs de paramètres remplacées pour déterminer si une pile a dérivé. Pour de plus amples informations sur le remplacement des paramètres de modèle dans les instances de

pile, veuillez consulter [Remplacez les valeurs des paramètres sur les piles de votre CloudFormation StackSet](#).

Si vous effectuez une dérive de [directement sur une pile](#) associée à une instance de pile, ces résultats de dérive ne sont pas disponibles sur la page de StackSetsconsole.

Détecter la dérive sur une StackSet (console)

Pour détecter une dérive sur un StackSet

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la StackSetspage, sélectionnez celui StackSet sur lequel vous souhaitez effectuer une détection de dérive.
3. Dans le menu Actions sélectionnez Detect drifts (Détecter les écarts).

CloudFormation affiche une barre d'information indiquant que la détection de dérive a été initiée pour le produit sélectionné StackSet.

4. Facultatif : pour suivre la progression de l'opération de détection de l'écart :
 - a. Sélectionnez le StackSet nom pour afficher la page de détails du Stackset.
 - b. Sélectionnez l'onglet Operations (Opérations) sélectionnez l'opération de détection de l'écart, puis sélectionnez View drift details (Afficher les détails de l'écart).

CloudFormation affiche la boîte de dialogue Détails de l'opération.

5. Attendez que l'opération de détection de dérive soit CloudFormation terminée. Lorsque l'opération de détection de dérive est terminée, l'état de dérive et l'heure de la dernière vérification de dérive sont mis à CloudFormation jour pour votre StackSet. Ces champs sont répertoriés dans l'onglet Vue d'ensemble de la page de StackSet détails de la sélection StackSet.

L'opération de détection de dérive peut prendre un certain temps, en fonction du nombre d'instances de pile incluses dans le StackSet et du nombre de ressources incluses dans le StackSet. Vous ne pouvez exécuter qu'une seule opération de détection de dérive sur une même StackSet machine à la fois. CloudFormation poursuit l'opération de détection de dérive même après avoir fermé la barre d'informations.

6. Pour consulter les résultats de détection de dérive pour les instances de pile d'un StackSet, sélectionnez l'onglet Stack instances.

La colonne Nom de la pile répertorie le nom de la pile associée à chaque instance de pile et la colonne Statut de l'écart répertorie le statut de l'écart de cette pile. Une pile est considérée comme ayant dérivé si une ou plusieurs de ses ressources ont dérivé.

7. Pour consulter les résultats de détection de l'écart pour la pile associée à une instance de pile spécifique :
 - a. Choisissez l'onglet Opérations.
 - b. Sélectionnez l'opération d'écart pour laquelle vous souhaitez afficher les résultats de détection de dérive l'écart. Un volet fractionné affichera le statut et le motif de l'instance de pile pour l'opération sélectionnée. Pour une opération d'écart, la colonne du motif du statut indique le statut de l'écart d'une instance de pile.
 - c. Choisissez l'instance de pile pour laquelle vous souhaitez afficher les détails de l'écart, puis choisissez Afficher les écarts des ressources. La table Statut de l'écart de ressource de la page Écarts de ressources répertorie chaque ressource de pile avec son statut d'écart et la dernière fois que la détection de l'écart a été lancée sur la ressource. L'ID logique et l'ID physique de chaque ressource sont affichés pour vous aider à l'identifier.
8. Vous pouvez trier les ressources en fonction de leur statut d'écart à l'aide de la colonne Statut de l'écart.

Pour afficher les détails sur une ressource modifiée :

- La ressource étant sélectionnée, sélectionnez Afficher les détails de l'écart.

CloudFormation affiche la page détaillée de dérive pour cette ressource en particulier. Cette page répertorie les différences entre les ressources. Elle répertorie également les valeurs des propriétés attendues et actuelles des ressources.

Note

Si la pile appartient à une région et à un compte différents de ceux auxquels vous êtes actuellement connecté, le bouton Détecter l'écart sera désactivé et vous ne pourrez pas afficher les détails.

Détecter la dérive sur un StackSet (AWS CLI)

Pour détecter la dérive sur l'ensemble d'une pile à l'aide du AWS CLI, procédez comme suit :

Pour détecter une dérive sur un StackSet

1. Utilisez la [detect-stack-set-drift](#) commande pour détecter la dérive sur l'ensemble d'une pile StackSet et sur ses instances de pile associées.

L'exemple suivant lance la détection de dérive sur. StackSet `stack-set-drift-example`

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

Sortie :

```
{  
  "OperationId": "c36e44aa-3a83-411a-b503-cb611example"  
}
```

2. Les opérations de détection de StackSet dérive pouvant être une opération de longue durée, utilisez la [describe-stack-set-operation](#) commande pour surveiller l'état de l'opération de dérive. Cette commande prend l'ID d' StackSet opération renvoyé par la `detect-stack-set-drift` commande.

Les exemples suivants utilisent l'ID d'opération de l'exemple précédent pour renvoyer des informations sur l'opération de détection de StackSet dérive. Dans cet exemple, l'opération est toujours en cours d'exécution. Sur les sept instances de pile associées à cela StackSet, une instance de pile a déjà été détectée comme étant dérivée, deux instances sont synchronisées et la détection de la dérive pour les quatre instances de pile restantes est toujours en cours. Comme une instance a dérivé, le statut de dérive de l'instance StackSet elle-même est maintenant `DRIFTED`.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Sortie :

```
{  
  "StackSetOperation": {  
    "Status": "RUNNING",  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
    AWSCloudFormationStackSetAdministrationRole",  
  }  
}
```

```

    "OperationPreferences": {
      "RegionOrder": []
    },
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "StackSetDriftDetectionDetails": {
      "DriftedStackInstancesCount": 1,
      "TotalStackInstancesCount": 7,
      "LastDriftCheckTimestamp": "2019-12-04T20:34:28.543Z",
      "InSyncStackInstancesCount": 2,
      "InProgressStackInstancesCount": 4,
      "DriftStatus": "DRIFTED",
      "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}

```

Cet exemple illustre la même commande exécutée ultérieurement et montre les informations renvoyées une fois l'opération de détection de l'écart terminée. Deux des sept instances de pile associées à cela StackSet ont dérivé, ce qui donne à l'état de dérive de l'instance StackSet elle-même la forme DRIFTED suivante.

```

aws cloudformation describe-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id c36e44aa-3a83-411a-b503-cb611example

```

Sortie :

```

{
  "StackSetOperation": {
    "Status": "SUCCEEDED",
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "OperationPreferences": {
      "RegionOrder": []
    }
  },
  "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
  "EndTimestamp": "2019-12-04T20:37:32.829Z",

```



```

    "StackSetDriftDetectionDetails": {
      "DriftedStackInstancesCount": 2,
      "TotalStackInstancesCount": 7,
      "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
      "InSyncStackInstancesCount": 5,
      "InProgressStackInstancesCount": 0,
      "DriftStatus": "DRIFTED",
      "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}

```

3. Lorsque l'opération de détection de StackSet dérive est terminée, utilisez les commandes `list-stack-instance-resource-drifts`, `list-stack-instances`, `describe-stack-set`, `describe-stack-instance`, et `describe-stack-set-operation` pour examiner les résultats.

La commande [describe-stack-set](#) inclut les mêmes informations détaillées sur l'écart que celles renvoyées par la commande `describe-stack-set-operation`.

```

aws cloudformation describe-stack-set \
  --stack-set-name stack-set-drift-example

```

Sortie :

```

{
  "StackSet": {
    "Status": "ACTIVE",
    "Description": "Demonstration of drift detection on stack sets.",
    "Parameters": [],
    "Tags": [
      {
        "Value": "Drift detection",
        "Key": "Feature"
      }
    ],
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "Capabilities": [],

```

```

    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "StackSetDriftDetectionDetails": {
        "DriftedStackInstancesCount": 2,
        "TotalStackInstancesCount": 7,
        "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
        "InProgressStackInstancesCount": 0,
        "DriftStatus": "DRIFTED",
        "DriftDetectionStatus": "COMPLETED",
        "InSyncStackInstancesCount": 5,
        "FailedStackInstancesCount": 0
    },
    "StackSetARN": "arn:aws:cloudformation:us-east-1:123456789012:stackset/
stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",
    "TemplateBody": [details omitted],
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22ebexample",
    "StackSetName": "stack-set-drift-example"
}
}

```

Vous pouvez utiliser la [list-stack-instances](#) commande pour renvoyer des informations récapitulatives sur les instances de pile associées à a StackSet, y compris l'état de dérive de chaque instance de pile.

Dans cet exemple, l'exécution list-stack-instances sur l'exemple StackSet avec le filtre d'état de dérive défini sur vous DRIFTED permet d'identifier les deux instances de pile ayant un statut de dérive de DRIFTED.

```

aws cloudformation list-stack-instances \
  --stack-set-name stack-set-drift-example \
  --filters Name=DRIFT_STATUS,Values=DRIFTED

```

Sortie :

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
StackSet-stack-set-drift-example-b0fb6083-60c0-4e39-
af15-2f071e0db90c/0e4f0940-16d4-11ea-93d8-0641cexample",
      "Status": "CURRENT",

```

```

        "Account": "012345678910",
        "Region": "eu-west-1",
        "LastDriftCheckTimestamp": "2019-12-04T20:37:32.687Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
        "Status": "CURRENT",
        "Account": "123456789012",
        "Region": "us-east-1",
        "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    [additional stack instances omitted]
]
}

```

La commande [describe-stack-instance](#) renvoie également ces informations, mais pour une seule instance de pile, comme dans l'exemple ci-dessous.

```

aws cloudformation describe-stack-instance \
  --stack-set-name stack-set-drift-example \
  --stack-instance-account 012345678910 --stack-instance-region us-east-1

```

Sortie :

```

{
  "StackInstance": {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
    "Status": "CURRENT",

```

```

    "Account": "123456789012",
    "Region": "us-east-1",
    "ParameterOverrides": [],
    "DriftStatus": "DRIFTED",
    "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
    "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}

```

- Une fois que vous avez identifié les instances de pile qui ont dérivé, vous pouvez utiliser les informations sur les instances de pile renvoyées par les commandes `list-stack-instances` ou `describe-stack-instance` pour exécuter la commande [list-stack-instance-resource-drifts](#). Cette commande renvoie des informations détaillées sur les ressources de la pile qui présentent un écart pour une opération d'écart particulière.

L'exemple suivant utilise le paramètre `--stack-instance-resource-drift-statuses` pour demander des informations d'écart de pile pour les ressources qui ont été modifiées ou supprimées dans le précédent exemple d'opération d'écart. La demande renvoie des informations sur la seule ressource qui a été modifiée, y compris des détails sur deux de ses propriétés et leurs valeurs qui ont été modifiées. Aucune ressource n'a été supprimée.

```

aws cloudformation list-stack-instance-resource-drifts \
  --stack-set-name my-stack-set-with-resource-drift \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-east-1 \
  --operation-id c36e44aa-3a83-411a-b503-cb611example \
  --stack-instance-resource-drift-statuses MODIFIED DELETED

```

Sortie :

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-
stack-set-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ResourceType": "AWS::SQS::Queue",
      "Timestamp": "2018-03-26T17:23:34.489Z",
      "PhysicalResourceId": "https://sqs.us-
east-1.amazonaws.com/123456789012/my-stack-with-resource-drift-Queue-494PBHC076H4",

```

```
    "StackResourceDriftStatus": "MODIFIED",
    "PropertyDifferences": [
      {
        "PropertyPath": "/DelaySeconds",
        "ActualValue": "120",
        "ExpectedValue": "20",
        "DifferenceType": "NOT_EQUAL"
      },
      {
        "PropertyPath": "/RedrivePolicy/maxReceiveCount",
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
      }
    ],
    "LogicalResourceId": "Queue"
  }
]
```

Arrêter la détection de dérive sur un StackSet

Comme la détection de dérive sur un StackSet peut être une opération de longue durée, il peut arriver que vous souhaitiez arrêter une opération de détection de dérive en cours d'exécution sur un StackSet.

Pour arrêter la détection de dérive sur une StackSet (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Sur la StackSetspage, sélectionnez le nom du StackSet.

CloudFormation affiche la page de StackSets détails de la sélection StackSet.

3. Sur la page de StackSets détails, sélectionnez l'onglet Opérations, puis sélectionnez l'opération de détection de dérive.
4. Sélectionnez Stop operation (Arrêter l'opération).

Pour arrêter la détection de dérive sur un StackSet (AWS CLI)

- Utilisez la commande [stop-stack-set-operation](#). Vous devez fournir à la fois le StackSet nom et l'identifiant de l'opération de détection de StackSet dérive.

```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id 624af370-311a-11e8-b6b7-500cexample
```

Importez des piles dans CloudFormation StackSets

Une opération d'importation de piles peut importer des piles existantes dans des piles nouvelles ou existantes StackSets, afin que vous puissiez migrer des piles existantes vers une opération StackSet en une seule opération.

Pour l'autogestion StackSets, l'opération d'importation peut importer des piles dans le compte administrateur ou dans différents comptes cibles et. Régions AWS Pour les services gérés StackSets, l'opération d'importation peut importer n'importe quelle pile au même AWS Organizations titre que le compte de gestion.

Voici les considérations et limites à prendre en compte lors de l'importation de piles dans StackSets :

- L'opération d'importation peut importer jusqu'à 10 piles à l'aide d'une pile intégrée IDs ou jusqu'à 200 piles à l'aide d'un objet Amazon S3.
- La propriété NoEcho n'est pas prise en charge. Les piles qui en contiennent NoEcho ne seront pas importées dans un nouveau StackSets par le biais de l' StackSet importation.
- Les piles ne peuvent appartenir qu'à un seul StackSet.
- Vous pouvez implémenter des balises de pile dans le StackSet en spécifiant explicitement des balises en tant que paramètres lors de l'opération d'importation de pile.
- Les remplacements de paramètres personnalisés d'une pile ne sont pas affectés lors de l'opération d'importation.
- Les paramètres des instances de stack importées ne peuvent pas être mis à jour à l'aide de l'option Modifier StackSet les détails. Vous devez utiliser les StackSet paramètres Override. Pour plus d'informations sur le remplacement des paramètres, voir [Remplacer les paramètres sur les piles](#).
- Les StackSets quotas et les instances de pile s'appliquent lors de l'importation de piles. Pour plus d'informations sur les quotas, consultez [Comprendre les CloudFormation quotas](#).

Rubriques

- [Importation de stack autogérée pour CloudFormation StackSets](#)
- [Importation de stack gérée par des services pour CloudFormation StackSets](#)
- [Rétablir les importations de piles dans CloudFormation StackSets](#)

Importation de stack autogérée pour CloudFormation StackSets

L'opération d'importation de CloudFormation piles peut importer des piles existantes dans des piles nouvelles ou existantes StackSets, afin que vous puissiez migrer des piles existantes vers une opération StackSet en une seule opération. En utilisant l'importation de pile, vous évitez les temps d'arrêt et les pannes sans supprimer et recréer ces ressources. Une fois la pile importée dans un StackSet, la pile d'origine devient une instance de pile de l'ensemble de piles spécifié.

Considérations relatives à l'importation de piles autogérées

- L'opération d'importation de piles nécessite un compte administrateur dans lequel vous créez un compte StackSet et un compte cible contenant une pile.
- Le compte de destination doit être autorisé à utiliser l'opération `GetTemplate` avec l'entrée de l'ID de pile ou de l'ARN. Pour cette raison, votre compte administrateur doit être `AWSCloudFormationStackSetsExecutionRole` autorisé `AWSCloudFormationStackSetAdministrationRole`.

Rubriques

- [Importer une pile existante dans une nouvelle StackSet \(console\)](#)
- [Importer une pile existante dans une pile existante StackSet \(console\)](#)
- [Importer une pile dans un StackSet \(AWS CLI\)](#)

Importer une pile existante dans une nouvelle StackSet (console)

Avant de commencer, identifiez la pile que vous voulez importer.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez StackSets.
3. En haut de la StackSetspage, choisissez Create StackSet.

4. Sur la page Choisir un modèle, spécifiez un modèle à l'aide de l'une des options suivantes et sélectionnez Suivant.
 - Choisissez Amazon S3 URL (URL Amazon S3), puis spécifiez l'URL de votre modèle dans la zone de texte.
 - Choisissez Upload a template file (Charger un fichier de modèle), puis recherchez votre modèle.
 - Choisissez From stack ID (Depuis l'ID de la pile) et saisissez l'ID de votre pile.
5. Sur la page Spécifier StackSet les détails, entrez le nom du StackSet fichier que vous souhaitez créer et choisissez Suivant.

(Facultatif) Entrez une description du StackSet.

6. Sur la page des StackSet options de configuration, passez en revue vos choix et choisissez Suivant.
7. Sur la page Définir les options de déploiement, sélectionnez Importer des piles dans le StackSet.
8. Saisissez l'ID de la pile que vous souhaitez importer dans le champ Stacks to import (Piles à importer). Par exemple, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Facultatif) Sélectionnez Ajouter un autre ID de pile et saisissez l'ID d'une autre pile que vous voulez importer. Vous pouvez ajouter jusqu'à 10 piles par opération d'importation de pile.

9. Vérifiez vos options de déploiement et sélectionnez Suivant.
10. Sur la page Révision, passez en revue vos choix et vos StackSet propriétés. Lorsque vous êtes prêt à importer votre pile dans votre StackSet, choisissez Soumettre.

Résultats : La pile importée est désormais une instance de pile de la pile spécifiée StackSet. Pour en savoir plus sur l'état de l'importation de la pile, consultez [StackSets codes de statut](#).

Importer une pile existante dans une pile existante StackSet (console)

Avant de commencer, identifiez la pile que vous voulez importer.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez StackSets.
3. Sur la StackSetspage, choisissez StackSet celui dans lequel vous souhaitez importer une pile.

4. Lorsque cette option est StackSet sélectionnée, choisissez Ajouter des piles à dans le StackSet menu Actions.
5. Sur la page Définir les options de déploiement, sélectionnez Importer des piles dans le StackSet et saisissez l'ID de la pile que vous voulez importer dans le champ Piles à importer. Par exemple, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Facultatif) Sélectionnez Ajouter un autre ID de pile et saisissez l'ID d'une autre pile que vous voulez importer. Vous pouvez ajouter jusqu'à 10 piles par opération d'importation de pile.
6. Choisissez Suivant.
7. Sur la page Spécifier les remplacements, vérifiez vos choix et sélectionnez Suivant.
8. Sur la page Révision, passez en revue vos choix et vos StackSet propriétés. Lorsque vous êtes prêt à créer votre StackSet, choisissez Soumettre.

Résultats : La pile importée est désormais une instance de pile de la pile spécifiée StackSet. Pour en savoir plus sur l'état de l'importation de la pile, consultez [StackSets codes de statut](#).

Importer une pile dans un StackSet (AWS CLI)

Pour importer une pile existante dans une nouvelle StackSet

La `create-stack-set` commande suivante crée StackSet et importe la pile spécifiée. La pile à importer est identifiée par son ARN. Remplacez chaque espace réservé par vos propres informations.

```
aws cloudformation create-stack-set \  
  --stack-set-name MyStackSet \  
  --stack-id arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/466df9e0-0dff-08e3-8e2f-5088487c4896 \  
  --administration-role-arn arn:aws:iam::123456789012:role/AWSCloudFormationStackSetAdministrationRole \  
  --execution-role-name AWSCloudFormationStackSetExecutionRole
```

Pour importer une pile existante dans une pile existante StackSet

La `import-stacks-to-stack-sets` commande suivante importe la pile spécifiée dans le *MyStackSet* StackSet. La pile à importer est identifiée par son ARN. Remplacez chaque espace réservé par vos propres informations.

```
aws cloudformation import-stacks-to-stack-set \  

```

```
--stack-set MyStackSet \  
--stack-ids arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/  
f449b250-b969-11e0-a185-5081d0136786
```

Pour spécifier plusieurs piles, veuillez utiliser le format suivant pour la valeur de l'option `--stack-ids`.

```
--stack-ids "arn_1" "arn_2"
```

Pour cloner la pile importée dans d'autres régions et comptes

La `create-stack-instances` commande suivante ajoute des instances de pile à votre StackSet. Remplacez chaque espace réservé par vos propres informations.

```
aws cloudformation create-stack-instances \  
--stack-set-name MyStackSet \  
--accounts '["account_ID_1", "account_ID_2"]' \  
--regions '["region_1", "region_2"]'
```

Importation de stack gérée par des services pour CloudFormation StackSets

L'opération d'importation de CloudFormation piles peut importer des piles existantes dans des piles nouvelles ou existantes StackSets, afin que vous puissiez migrer des piles existantes vers une opération StackSet en une seule opération. StackSets étend les fonctionnalités des piles afin que vous puissiez créer, mettre à jour ou supprimer des piles sur plusieurs comptes et régions en une seule opération.

Considérations relatives aux importations de piles gérées par le service

- L'opération d'importation de piles nécessite un compte de gestion ou un compte administrateur délégué dans lequel vous pouvez gérer les éléments associés, AWS Organizations par exemple en activant un accès sécurisé avec StackSets.
- Les comptes cibles doivent être membres du compte AWS Organizations géré par le compte de gestion ou du compte administrateur délégué.
- La pile de cibles existe dans l'une des cibles OUs.
- Le compte cible doit être membre de AWS Organizations.

- AWS Organizations l'accès doit être conforme à l'ACTIVATED état pour les Organisations.
- Les piles importées doivent être présentes dans l'un des comptes membres, et non dans le compte de gestion.

Rubriques

- [Importer une pile gérée par des services dans une nouvelle StackSet \(console\)](#)
- [Création et importation d'une pile gérée par des services dans une pile existante StackSet \(console\)](#)
- [Importer une pile gérée par des services dans une pile existante StackSet \(console\)](#)
- [Importer une pile gérée par des services dans un StackSet \(\)AWS CLI](#)

Importer une pile gérée par des services dans une nouvelle StackSet (console)

Importez une pile dans une nouvelle StackSet en utilisant le AWS Management Console

Pour importer une nouvelle pile dans un StackSet, identifiez une pile contenant la ressource que vous souhaitez importer.

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez StackSets.
3. En haut de la StackSets page, choisissez Create StackSet.
4. Sur la page Choisir un modèle, procédez comme suit :
 - a. Pour le modèle StackSet d'autorisation, choisissez Autorisations gérées par le service.
 - b. Pour Prérequis – Préparer le modèle, sélectionnez Le modèle est prêt, puis choisissez votre modèle à l'aide de l'une des options suivantes :
 - Pour URL Amazon S3, saisissez votre URL Amazon S3 dans le champ URL Amazon S3.
 - Pour Télécharger un fichier modèle, choisissez un CloudFormation modèle sur votre ordinateur local.

Acceptez vos paramètres et choisissez Next (Suivant).

5. Sur la page Spécifier StackSet les détails, procédez comme suit :

- a. Entrez un StackSet nom dans la zone StackSet de nom.
- b. (Facultatif) Entrez une description dans la section de StackSet description.

Sur la page des StackSet options de configuration, passez en revue vos choix et choisissez Suivant.

6. Sur la page Définir les options de déploiement, procédez comme suit :
 - a. Pour Ajouter des piles au jeu de piles, sélectionnez Importer des piles dans le jeu de piles.
 - b. Pour Stacks to import (Piles à importer), choisissez la méthode d'importation de votre pile.
 - i. Pour Stack ID (ID de la pile), saisissez l'ID de votre pile.
 - ii. Pour URL de la pile, saisissez l'URL Amazon S3.
7. Sous Associer des unités organisationnelles, procédez comme suit :
 - a. Sélectionnez Associer à l'organisation pour utiliser l'unité organisationnelle racine.
 - b. Choisissez Associer aux unités organisationnelles (OUs) pour saisir l'unité d'organisation parent IDs pour les piles à importer. Par exemple, si Stack 1 et Stack 2 sont sous OU1, et si Stack 3 est sous OU2, saisissez OU1 et OU2.

Acceptez vos paramètres et choisissez Next (Suivant).

8. Sur la page Révision, passez en revue vos paramètres et choisissez Submit (Soumettre).

Création et importation d'une pile gérée par des services dans une pile existante StackSet (console)

Pour importer une pile existante dans une nouvelle StackSet, identifiez une pile contenant la ressource que vous souhaitez importer.

Pour créer StackSet et importer une pile

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans le volet de navigation, choisissez StackSets.
3. En haut de la StackSetspage, choisissez Create StackSet.
4. Sur la page Choisir un modèle, procédez comme suit :

- a. Pour le modèle StackSet d'autorisation, choisissez Autorisations gérées par le service.
- b. Pour Prérequis – Préparer le modèle, sélectionnez Le modèle est prêt, puis choisissez votre modèle à l'aide de l'une des options suivantes :
 - Pour URL Amazon S3, saisissez votre URL Amazon S3 dans le champ URL Amazon S3.
 - Pour Télécharger un fichier modèle, choisissez un CloudFormation modèle sur votre ordinateur local.

Acceptez vos paramètres et choisissez Next (Suivant).

5. Sur la page Spécifier StackSet les détails, procédez comme suit :

- a. Entrez un StackSet nom dans la zone StackSet de nom.
- b. (Facultatif) Entrez une description dans la section de StackSet description.

Sur la page des StackSet options de configuration, passez en revue vos choix et choisissez Suivant.

6. Sur la page Définir les options de déploiement, procédez comme suit :

- Pour Add stacks to stack set (Ajouter des piles à un ensemble de piles), sélectionnez Deploy new stacks (Déployer de nouvelles piles).

7. Dans la section Associer des unités organisationnelles, procédez comme suit :

- a. Sélectionnez Associer à l'organisation pour utiliser l'unité organisationnelle racine.
- b. Choisissez Associer aux unités organisationnelles (OUs) pour saisir l'unité d'organisation parent IDs pour les piles à importer. Par exemple, si Stack 1 et Stack 2 sont sous OU1, et si Stack 3 est sous OU2, saisissez OU1 et OU2.

8. Pour Specify regions (Spécifier des régions) et Deployment options (Options de déploiement), vérifiez vos choix.

Acceptez vos paramètres et choisissez Next (Suivant).

9. Sur la page Révision, passez en revue vos paramètres et choisissez Submit (Soumettre).

Importer une pile gérée par des services dans une pile existante StackSet (console)

Choisissez votre pile StackSet et identifiez la pile que vous souhaitez importer.

Pour importer une pile dans une pile existante StackSet

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
 2. Dans le volet de navigation, choisissez StackSets.
 3. Choisissez le StackSet fichier dans lequel vous souhaitez importer une pile, puis choisissez Ajouter des piles dans le StackSet menu déroulant Actions.
 4. Sur la page Définir les options de déploiement, procédez comme suit :
 - a. Pour Ajouter des piles au jeu de piles, sélectionnez Importer des piles dans le jeu de piles.
 - b. Sous Piles à importer, procédez comme suit
 - i. Pour Stack ID (ID de la pile), saisissez l'ID de votre pile.
 - ii. Pour URL de la pile, saisissez l'URL Amazon S3.
 - c. Sous Associer des unités organisationnelles, procédez comme suit :
 - i. Sélectionnez Associer à l'organisation pour utiliser l'unité organisationnelle racine.
 - ii. Choisissez Associer aux unités organisationnelles (OUs) pour saisir l'unité d'organisation parent IDs pour les piles à importer. Par exemple, si Stack 1 et Stack 2 sont sous OU1, et si Stack 3 est sous OU2, saisissez OU1 et OU2.
- Acceptez vos paramètres et choisissez Next (Suivant).
5. Vérifiez la page Spécifier des remplacements et choisissez Next (Suivant).
 6. Confirmez et examinez la page Révision et choisissez Submit (Soumettre).

Importer une pile gérée par des services dans un StackSet ()AWS CLI

Une fois qu'un StackSet est créé, vous pouvez importer vos piles en transmettant les ID de pile des piles importées. Vous pouvez également transmettre la liste d'ID d'unité d'organisation à laquelle vous souhaitez le mapper.

CloudFormation importera les piles fournies par l'utilisateur dans celles-ci OUs et les utilisera OUs comme cibles de déploiement pour le StackSet. La pile IDs présentée en entrée sera mappée à l'unité d'organisation la plus proche dans la liste d'identifiants d'unités d'organisation saisie en interne. Si une pile n'appartient pas à un ID d'unité existant dans la liste d'entrée, le StackNotFoundException message d'erreur AWS CLI sera renvoyé.

L'opération `import-stacks-to-stack-set` crée des instances de pile pour les piles dans l'entrée d'ID d'unité d'organisation. Les AWS CLI exemples suivants utilisent l'`import-stacks-to-stack-set` opération pour importer une pile dans un StackSet.

- Pour utiliser l'opération `import-stacks-to-stack-sets`, spécifiez les `stack-ids` ou `stack-ids-url` que vous souhaitez importer dans votre ensemble de piles.

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceManagedStackSet \  
  --stack-ids "arn:123456789012:us-east-1:Stack1" \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceManagedStackSet \  
  --stack-ids-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-name.json \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

Note

L'opération `import-stacks-to-stack-sets` vous oblige à spécifier au moins un ID d'unité d'organisation afin qu'elle puisse associer la pile importée à cette unité d'organisation particulière. Cette opération ne crée pas d'instances de pile pour les autres comptes membres dans le compte associé OUs. Pour mettre à jour les comptes des membres associés OUs, utilisez `create-stack-instances` ou `update-stack-instances`.

`create-stack-set` crée des instances de pile pour tous les comptes sous le OUs modèle fourni par l'utilisateur, soit depuis Direct Upload, soit depuis Amazon S3. Les AWS CLI exemples suivants utilisent l'`create-stack-set` opération pour importer une pile dans une nouvelle StackSet.

- Pour utiliser cette `create-stack-set` opération, spécifiez votre StackSet nom et importez une pile dans une pile nouvellement créée StackSet.

```
aws cloudformation create-stack-set \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-name.json \  
  --permission-model SERVICE_MANAGED \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
--auto-deployment Enabled=true
```

Rétablir les importations de piles dans CloudFormation StackSets

Si vous constatez des modifications indésirables dans votre instance de pile, vous pouvez revenir sur l'importation de l'instance de pile.

Lorsque vous annulez l'importation d'une instance de pile, elle CloudFormation est supprimée de la pile StackSet mais conserve les ressources de la pile.

Pour annuler une opération d'importation de pile, procédez comme suit.

1. Spécifiez un attribut `DeletionPolicy` de `Retain` pour chaque ressource que vous souhaitez conserver après la suppression de l'instance de pile. Pour de plus amples informations, veuillez consulter [Rétablissement d'une opération d'importation](#).
2. Supprimez les instances de pile de votre StackSet. Pour de plus amples informations, veuillez consulter [Supprimer des piles de CloudFormation StackSets](#).
3. Supprimez votre StackSet. Pour de plus amples informations, veuillez consulter [Supprimer CloudFormation StackSets](#).

Bonnes pratiques d'utilisation CloudFormation StackSets

Cette section décrit les meilleures pratiques pour définir un StackSet modèle, créer ou ajouter des piles à un StackSet, ou mettre à jour un StackSet.

Si vous êtes nouveau dans ce domaine CloudFormation, consultez la [CloudFormation meilleures pratiques](#) rubrique pour obtenir d'autres recommandations susceptibles de vous aider à l'utiliser de manière CloudFormation plus efficace et plus sûre.

Rubriques

- [Définition du modèle](#)
- [Création ou ajout de piles dans StackSet](#)
- [Mettre à jour les piles dans un StackSet](#)

Définition du modèle

- Définissez le modèle à normaliser dans les différents comptes et les différentes régions.
- Au fur et à mesure que vous créez le modèle, assurez-vous que les ressources globales (par exemple, les rôles IAM et les compartiments Amazon S3) n'entraînent pas de conflits de noms lorsqu'elles sont créées dans plusieurs régions dans le même compte.
- A StackSet possède un modèle et un ensemble de paramètres uniques. La même pile est créée dans tous les comptes associés à un StackSet. Au fur et à mesure que vous créez vos modèles, rendez-les suffisamment précis pour obtenir un bon équilibre entre le contrôle et la normalisation.
- Nous vous recommandons de stocker votre modèle dans un compartiment Amazon S3.

Création ou ajout de piles dans StackSet

- Vérifiez que l'ajout d'instances de pile à votre configuration initiale StackSet fonctionne avant d'ajouter un plus grand nombre d'instances de pile à votre StackSet.
- Choisissez les options de déploiement qui fonctionnent pour votre cas d'utilisation.
 - Pour effectuer un déploiement plus conservatif, définissez Maximum Concurrent Accounts (Nombre maximal de comptes en parallèle) sur 1 et Failure Tolerance (Tolérance aux pannes) sur 0. Placez votre région où l'impact reste mineur à la première place de la liste Region Order (Ordre des régions). Commencez avec une seule région.
 - Pour accélérer le déploiement, augmentez les valeurs de Maximum Concurrent Accounts (Nombre maximal de comptes en parallèle) et de Failure Tolerance (Tolérance aux pannes) selon vos besoins.
- Les opérations StackSets dépendent du nombre d'instances de stack impliquées et peuvent prendre beaucoup de temps.

Mettre à jour les piles dans un StackSet

- Par défaut, la mise à jour d'une StackSet met à jour toutes les instances de pile. Si vous avez 20 comptes chacun dans deux régions, vous aurez 40 instances de pile, et toutes seront mises à jour lorsque vous mettrez à jour le StackSet.

Dans le cas StackSets d'un grand nombre d'instances de pile, nous recommandons, pour tester la version mise à jour d'un modèle, de mettre à jour les instances de pile de manière sélective dans quelques comptes de test avant de mettre à jour toutes les instances de pile.

- Pour avoir un contrôle plus précis sur la mise à jour des piles individuelles au sein de votre entreprise StackSet, prévoyez d'en créer plusieurs. StackSets
- La mise à jour d'un fichier StackSet contenant un grand nombre de piles peut prendre beaucoup de temps. Dans cette version, une seule opération est autorisée à la fois sur un StackSet. Planifiez vos mises à jour de manière à ne pas être empêché d'effectuer d'autres opérations sur le StackSet.

CloudFormation StackSets exemples de modèles

Cette section contient des liens vers des exemples de CloudFormation modèles qui peuvent vous aider à CloudFormation StackSets les utiliser dans votre entreprise. Les modèles répertoriés dans cette section activent [AWS CloudTrail](#) ou [AWS Config](#) et les règles qu'il contient.

Important

Pour des raisons de sécurité, lorsque vous autorisez AWS Config l'accès à un compartiment Amazon S3, nous vous recommandons vivement de restreindre l'accès dans le cadre de la politique de compartiment à `AWS:SourceAccount` cette condition. Les nouveaux modèles ont été mis à jour pour inclure `AWS:SourceAccount`. Si votre politique de compartiment existante ne suit pas cette bonne pratique en matière de sécurité, nous vous recommandons vivement de la modifier pour inclure cette protection. Cela garantit que l'accès AWS Config est accordé au nom des utilisateurs attendus uniquement.

Description	Lien S3
Activer AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ActivatorAWSCloudtrail .yml
Activer AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ActivatorAWSConfig .yml
Activer AWS Config avec la journalisation centralisée	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ActivatorAWSConfig ForOrganizations .yml

Description	Lien S3
Activez les politiques par défaut d'Amazon Data Lifecycle Manager au AWS sein d'une organisation ou de AWS comptes spécifiques	- Politique par défaut pour les instantanés EBS — .yaml https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy - Politique par défaut pour les applications soutenues par EBS — AMIs Policy.yaml https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefault https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml
Configurer une AWS Config règle pour déterminer si elle CloudTrail est activée	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleCloudtrailEnabled.yml
Configurer une AWS Config règle pour déterminer si le MFA root est activé	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml
Configurer une AWS Config règle pour déterminer si EIPs elles sont jointes	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEipAttached.yml
Configurer une AWS Config règle pour déterminer si les volumes EBS sont chiffrés	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Résolution des problèmes CloudFormation StackSets

Cette rubrique présente certains StackSets problèmes courants et propose des solutions à ces problèmes.

Rubriques

- [Raisons les plus courantes de l'échec de l'opération de pile](#)
- [Nouvelle tentative non réussie de création ou mise à jour de pile](#)
- [Echec de la suppression de l'instance de pile](#)
- [L'opération d'importation de pile échoue](#)
- [Nombre de défaillances des instances de Stack pour les StackSets opérations](#)

Raisons les plus courantes de l'échec de l'opération de pile

Problème : Une opération de pile a échoué et l'instance de pile a le statut OUTDATED.

Cause : Plusieurs causes courantes sont associées à l'échec d'une opération de pile.

- Autorisations insuffisantes dans un compte de destination pour la création des ressources spécifiées dans votre modèle.
- Le CloudFormation modèle peut contenir des erreurs. Validez le modèle CloudFormation et corrigez les erreurs avant d'essayer de créer votre StackSet.
- Le modèle effectue peut-être une tentative de création de ressources globales qui doivent être uniques et qui ne le sont, par exemple des compartiments S3.
- Un numéro de compte de destination spécifié n'existe pas. Vérifiez les numéros de compte de destination que vous avez spécifiés sur la page Set deployment options (Définir les options de déploiement) de l'assistant.
- Le compte d'administrateur de compte n'a pas de relation d'approbation avec le compte de destination.
- Le nombre maximal d'une ressource spécifiée dans votre modèle est déjà atteint dans votre compte de destination. Par exemple, vous avez peut-être atteint le nombre limite de rôles IAM autorisés dans un compte de destination, mais le modèle crée d'autres rôles IAM.
- Vous avez atteint le nombre maximum de piles autorisées dans un StackSet. Pour le nombre maximum de piles par StackSet, voir [Comprendre les CloudFormation quotas](#).

Solution : Pour plus d'informations sur les autorisations requises pour les comptes cible et administrateur avant de pouvoir créer StackSets, consultez [Autorisation accordée à tous les utilisateurs du compte administrateur de gérer les piles dans tous les comptes cibles](#).

Nouvelle tentative non réussie de création ou mise à jour de pile

Problème : Une création ou mise à jour de pile a échoué, et l'instance de pile a le statut OUTDATED. Pour résoudre les problèmes liés à l'échec de la création ou de la mise à jour d'une pile, ouvrez la CloudFormation console et consultez les événements de la pile, dont le statut sera DELETED (en cas d'échec des opérations de création) ou FAILED (en cas d'échec des opérations de mise à jour). Dans les événements de la pile, recherchez la colonne Motif du statut. La valeur de Motif du statut explique pourquoi l'opération de pile a échoué.

Une fois que vous éliminé la cause sous-jacente de l'échec de la création de la pile et que vous êtes prêt à réessayer de créer la pile, effectuez les étapes suivantes.

Solution : Effectuez les étapes suivantes pour recommencer votre opération de pile.

1. Dans la console, sélectionnez celui StackSet qui contient la pile sur laquelle l'opération a échoué.
2. Dans le menu Actions, choisissez Modifier les StackSet détails pour réessayer de créer ou de mettre à jour des piles.
3. Sur la page Specify template (Spécifier un modèle), pour utiliser le même modèle CloudFormation, conservez l'option par défaut, Utiliser modèle en cours. Si votre opération de pile a échoué car le modèle nécessitait des modifications, et que vous souhaitez charger un modèle révisé, choisissez Upload a template to Amazon S3 (Télécharger un modèle sur Amazon S3) à la place, puis Browse (Parcourir) pour sélectionner votre modèle mis à jour. Lorsque vous avez fini de charger votre modèle révisé, cliquez sur Next (Suivant).
4. Sur la page Spécifier les détails de la pile, si vous ne modifiez aucun paramètre spécifique à votre modèle, sélectionnez Suivant.
5. Sur la page Set deployment options (Définir les options de déploiement), modifiez les valeurs par défaut de Maximum concurrent accounts (Nombre maximal de comptes en parallèle) et Failure tolerance (Tolérance aux pannes) si vous le souhaitez. Pour plus d'informations sur ces paramètres, consultez la page [StackSet options de fonctionnement](#).
6. Sur la page Vérification, vérifiez vos sélections et cochez la case pour reconnaître les fonctions IAM requises. Sélectionnez Soumettre.

7. Si votre pile n'est pas mise à jour avec succès, répétez cette procédure après avoir résolu les problèmes sous-jacents qui empêchent la création de la pile.

Echec de la suppression de l'instance de pile

Problème : échec de la suppression d'une pile.

Cause : la suppression échoue pour n'importe quelle pile dont la protection contre la résiliation a été activée.

Solution : déterminez si la protection contre la résiliation a été activée pour la pile. Si c'est le cas, désactivez la protection contre la résiliation, puis effectuez à nouveau l'opération de suppression de l'instance de pile.

L'opération d'importation de pile échoue

Problème : une opération d'importation de piles ne parvient pas à importer des piles existantes dans des piles nouvelles ou existantes StackSets. L'instance de pile se trouve à l'état INOPERABLE.

Solution : annulez l'opération d'importation de pile en procédant comme suit.

1. Utilisez l' `StackSetOption Supprimer les piles depuis` et activez-la `RetainStacks` lors de la configuration, puis supprimez les instances de pile de votre StackSet. Pour plus d'informations, consultez [Supprimer des piles de CloudFormation StackSets](#).
2. Vous verrez que les instances de pile StackSet sont mises à jour pour supprimer l'instance de INOPERABLE pile.
3. Corrigez les instances de pile en fonction de l'erreur d'échec de l'importation et recommencez l'opération d'importation de la pile.

Nombre de défaillances des instances de Stack pour les StackSets opérations

Le nombre d'instances de pile défaillantes vous avertit en cas d'échec de la mise en service ou la mise à jour des instances de pile. Ces instances de pile n'ont pas été déployées pour une ou plusieurs des raisons suivantes :

- Ressource(s) existante(s) dotée(s) d'une configuration similaire

- Dépendances manquantes, telles que les Gestion des identités et des accès AWS rôles (IAM)
- Autres facteurs de conflit

Si vous souhaitez effectuer un déploiement avec une simultanée maximale, la valeur maximale de simultanée est, au maximum, supérieure d'une unité à la valeur de tolérance aux pannes. Par exemple, si la valeur de tolérance aux pannes est de 9, la valeur maximale de simultanée ne peut pas être supérieure à 10. Ainsi l'opération renvoie SUCCEEDED même si certaines instances de pile ne sont pas mises à jour. Le nouveau nombre d'instances de pile défaillantes vous permet de déterminer si l'opération n'a abouti que de manière conditionnelle parce que la valeur de tolérances aux pannes est définie pour autoriser toutes les défaillances.

Vous pouvez utiliser le AWS Management Console AWS SDK ou AWS CLI pour obtenir le nombre d'échecs et filtrer les instances de pile afin de déterminer quelles instances doivent être redéployées.

Utilisation de la console

Pour afficher le nombre d'instances de pile défaillantes :

1. Ouvrez la [CloudFormationconsole](#) et choisissez StackSets.
2. Choisissez votre StackSet, puis cliquez sur l'onglet Opérations.
3. Sélectionnez un état dans la colonne Status (État) pour afficher les détails relatifs à l'état. Vous trouverez le nombre d'instances de pile défaillantes d'une opération particulière dans les détails relatifs à l'état.

Pour consulter le nombre, la région et l'état des instances de pile associées à l'opération :

1. Dans les détails relatifs à l'état, sélectionnez le nombre d'instances de pile défaillantes. Exemple : Instances de pile : **<number of failed stack instances>**.
2. Développez le panneau latéral en sélectionnant l'en-tête de celui-ci. Les résultats affichés dans le panneau latéral indiquent l'état des instances de pile une fois l'opération sélectionnée terminée.

Pour afficher les détails de l'instance de pile actuelle d'une opération :

1. Sélectionnez l'onglet Stack Instances (Instances de pile).
2. Utilisez le filtre suivant : Last operation ID (ID de la dernière opération). Les résultats indiquent l'état actuel et le motif de cet état depuis la dernière opération de modification de l'instance. Vous

pouvez utiliser ce filtre en combinaison avec compte AWS , région AWS , état détaillé, et état d'écart pour affiner davantage vos résultats de recherche.

À l'aide du AWS CLI

Pour connaître le nombre d'instances de pile défailtantes, appelez `describe-stack-set-operation` ou `list-stack-set-operations` et consultez `StatusDetails`.

```
aws cloudformation describe-stack-set-operation --stack-set-name ss1 \  
--operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60
```

```
{  
  "StackSetOperation": {  
    "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",  
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",  
    "Action": "CREATE",  
    "Status": "SUCCEEDED",  
    "OperationPreferences": {  
      "RegionOrder": [],  
      "FailureToleranceCount": 10,  
      "MaxConcurrentCount": 10  
    },  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",  
    "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",  
    "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",  
    "StatusDetails": {  
      "FailedStackInstancesCount": 3  
    }  
  }  
}
```

```
aws cloudformation list-stack-set-operations --stack-set-name ss1
```

```
{  
  "Summaries": [  
    {  
      "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",  
      "Action": "CREATE",  

```



```

    "Status": "SUCCEEDED",
    "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
    "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
    "StatusDetails": {
      "FailedStackInstancesCount": 3
    },
    "OperationPreferences": {
      "RegionOrder": [],
      "FailureToleranceCount": 10,
      "MaxConcurrentCount": 10
    }
  }
]
}

```

Pour obtenir un aperçu de l'historique d'une opération particulière, utilisez `list-stack-set-operation-results` afin d'afficher l'état et le motif de l'état de chaque instance de pile au terme de l'opération. Consultez l'exemple suivant pour trouver l'état (`Status`) et le motif de l'état (`StatusReason`) :

```

aws cloudformation list-stack-set-operation-results --stack-set-name ss1 \
  --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60 --
  filters Name=OPERATION_RESULT_STATUS,Values=FAILED

```

```

{
  "Summaries": [
    {
      "Account": "123456789012",
      "Region": "us-west-2",
      "Status": "FAILED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "AccountGateResult": {
        "Status": "SKIPPED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
      },
      "OrganizationalUnitId": ""
    },
    {

```

```

        "Account": "123456789012",
        "Region": "us-west-1",
        "Status": "FAILED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "AccountGateResult": {
            "Status": "SKIPPED",
            "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
        },
        "OrganizationalUnitId": ""
    },
    {
        "Account": "123456789012",
        "Region": "us-east-1",
        "Status": "FAILED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "AccountGateResult": {
            "Status": "SKIPPED",
            "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
        },
        "OrganizationalUnitId": ""
    }
]
}

```

Utilisez `list-stack-instances` avec les filtres `DETAILED_STATUS` et `LAST_OPERATION_ID` pour obtenir la liste des instances de pile qui ont échoué lors de la dernière opération au cours de laquelle un déploiement de celles-ci a été tenté. Voir l'indicateur `--filters` dans l'exemple comportant `DETAILED_STATUS` et `LAST_OPERATION_ID` :

```

aws cloudformation list-stack-instances --stack-set-name ss1 \
  --filters Name=DETAILED_STATUS,Values=FAILED Name=LAST_OPERATION_ID,Values=5550e62f-
c822-4331-88fa-21c1d7bafc60

```

```
{
```

```

"Summaries": [
  {
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Region": "us-east-1",
    "Account": "123456789012",
    "Status": "OUTDATED",
    "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
    "StackInstanceStatus": {
      "DetailedStatus": "FAILED"
    },
    "OrganizationalUnitId": "",
    "DriftStatus": "NOT_CHECKED",
    "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
  },
  {
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Region": "us-west-1",
    "Account": "123456789012",
    "Status": "OUTDATED",
    "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
    "StackInstanceStatus": {
      "DetailedStatus": "FAILED"
    },
    "OrganizationalUnitId": "",
    "DriftStatus": "NOT_CHECKED",
    "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
  },
  {
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Region": "us-west-2",
    "Account": "123456789012",
    "Status": "OUTDATED",
    "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
    "StackInstanceStatus": {
      "DetailedStatus": "FAILED"
    },
    "OrganizationalUnitId": "",
    "DriftStatus": "NOT_CHECKED",

```

```

        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    }
}

```

Pour trouver l'ID de la dernière opération afin de modifier une instance de pile, utilisez `list-stack-instances` ou `describe-stack-instance` de manière à obtenir `LastOperationId` :

```

aws cloudformation describe-stack-instance --stack-set-name ss1 \
  --stack-instance-account 123456789012 --stack-instance-region us-east-2

```

```

{
  "StackInstance": {
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Region": "us-west-2",
    "Account": "123456789012",
    "ParameterOverrides": [],
    "Status": "OUTDATED",
    "StackInstanceStatus": {
      "DetailedStatus": "FAILED"
    },
    "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
    "OrganizationalUnitId": "",
    "DriftStatus": "NOT_CHECKED",
    "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
  }
}

```

Synchronisation des piles avec le code source stocké dans un référentiel Git avec synchronisation depuis Git

Avec Git sync, vous pouvez gérer vos CloudFormation piles à l'aide du contrôle de source. Pour ce faire, configurez CloudFormation pour surveiller un dépôt Git. Le référentiel est surveillé pour détecter les modifications apportées à deux fichiers :

- Un fichier CloudFormation modèle qui définit une pile
- Un fichier de déploiement de pile contenant les paramètres qui configurent la pile

Avec Git sync, vous pouvez utiliser les pull requests et le suivi des versions pour configurer, déployer et mettre à jour vos CloudFormation stacks à partir d'un emplacement centralisé. Lorsque vous validez des modifications apportées au modèle ou au fichier de déploiement, la pile est CloudFormation automatiquement mise à jour. Si vous utilisez des pull requests, vous CloudFormation pouvez laisser un commentaire sur la pull request expliquant les modifications qui seront apportées à votre stack avant de la mettre à jour. Cependant, vous devez d'abord activer cette caractéristique.

Git sync fournit une interface de console que vous pouvez utiliser pour créer un lien vers un dépôt, générer un fichier de déploiement de stack, mettre à jour un CloudFormation modèle et envoyer une pull request à votre dépôt. La synchronisation Git fournit également un tableau de bord d'état que vous pouvez utiliser pour surveiller, modifier et dépanner les déploiements actifs de la pile de synchronisation Git. Git Sync est accessible via la [CloudFormation console](#) lorsque vous [créez une pile](#). Vous pouvez également accéder à Git Sync à l'aide de CodeConnections. Pour plus d'informations, consultez [Utilisation des configurations de synchronisation pour les référentiels liés](#) dans le Guide de l'utilisateur de la console des outils pour développeurs.

Git Sync prend [GitHub](#) en charge les référentiels [GitHub Enterprise](#), [GitLab](#), [Bitbucket](#) et [GitLab autogérés](#).

Note

La synchronisation Git est disponible dans les régions suivantes : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Californie du Nord), USA Ouest (Oregon), Canada (Centre), Asie-Pacifique (Mumbai), Asie-Pacifique (Tokyo), Asie-Pacifique (Séoul), Asie-Pacifique

(Singapour), Asie-Pacifique (Sydney), Europe (Irlande), Europe (Londres), Europe (Paris), Europe (Stockholm), Europe (Francfort), Europe (Milan) et Amérique du Sud (São Paulo).

Pour plus d'informations sur l'utilisation de la synchronisation depuis Git dans le cadre d'une stratégie multi-comptes, consultez l'article de blog intitulé [Utilisation de la synchronisation depuis Git de CloudFormation pour configurer des ressources dans des comptes clients](#).

Rubriques

- [Comment fonctionne Git Sync avec CloudFormation](#)
- [Conditions préalables à la synchronisation des piles vers un référentiel Git à l'aide de la synchronisation Git](#)
- [Créez une pile à partir du code source du référentiel avec la synchronisation Git](#)
- [Permet CloudFormation de publier un résumé des modifications de la pile dans les pull requests](#)
- [Tableau de bord d'état de la synchronisation Git](#)

Comment fonctionne Git Sync avec CloudFormation

Cette rubrique décrit le fonctionnement de la synchronisation Git et présente les concepts clés nécessaires à son utilisation.

Rubriques

- [Comment fonctionne la synchronisation Git](#)
- [Commentaires sur les demandes d'extraction](#)
- [Fichier de déploiement de pile](#)
- [CloudFormation fichier modèle](#)
- [Référentiel de définition du modèle](#)

Comment fonctionne la synchronisation Git

Pour utiliser Git sync, vous devez d'abord connecter un fournisseur Git à CloudFormation l'utilisation du [CodeConnections](#) service. Dans les procédures décrites dans ce guide, la connexion est créée via la CodeConnections console. Vous pouvez également créer la connexion avec le AWS CLI. Vous pouvez utiliser l'un des fournisseurs Git suivants :

- [GitHub](#)
- [GitHub Entreprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLabautogéré](#)

Ensuite, vous créez un CloudFormation modèle qui définit votre pile et vous l'ajoutez à votre référentiel. Ce fichier modèle est surveillé. CloudFormation met automatiquement à jour la pile lorsque des modifications y sont apportées.

Dans la CloudFormation console, vous créez une nouvelle pile et vous choisissez Sync from Git CloudFormation pour indiquer d'utiliser Git sync. Vous allez spécifier le dépôt et la branche que vous CloudFormation souhaitez surveiller, ainsi que le CloudFormation modèle de votre référentiel qui définit la pile.

Lors de la configuration, vous pouvez soit fournir votre propre fichier de déploiement de pile à partir de votre référentiel, soit demander à la synchronisation Git d'en générer un pour vous. Le fichier de déploiement de la pile contient des paramètres et des valeurs qui configurent les ressources de votre pile. Ce fichier de déploiement de stack est surveillé. CloudFormation met automatiquement à jour la pile lorsque des modifications y sont apportées.

Git sync crée une pull request dans votre dépôt pour synchroniser votre pile avec le fichier CloudFormation modèle et le fichier de déploiement de la pile. Si c'est la synchronisation Git qui génère le fichier de déploiement de la pile pour vous, ce fichier est envoyé à votre référentiel par la synchronisation Git.

Vous fusionnez ensuite la pull request avec votre référentiel afin que celui-ci CloudFormation approvisionne la pile, la configure avec vos paramètres de déploiement et commence à surveiller les modifications apportées à votre référentiel.

À partir de ce moment, chaque fois que vous apportez des modifications à votre fichier modèle ou à votre fichier de déploiement de pile et que vous les validez dans votre référentiel, les modifications CloudFormation seront automatiquement détectées. Si votre équipe utilise des requêtes pull, les membres de votre équipe peuvent alors examiner et approuver les modifications avant leur déploiement. Une fois la pull request acceptée, CloudFormation déploie vos modifications.

Vous pouvez surveiller l'état de votre configuration de synchronisation Git pour la pile et consulter l'historique des validations appliquées à la pile dans la CloudFormation console. La console fournit également des outils pour reconfigurer la synchronisation Git et résoudre les problèmes.

Commentaires sur les demandes d'extraction

Vous pouvez choisir de CloudFormation créer un résumé des modifications de code dans les pull requests via le CodeConnections service en activant l'option Activer les commentaires sur les pull requests dans la console. Fournir un résumé des modifications apportées dans les demandes d'extraction permet aux membres de l'équipe d'examiner et de comprendre facilement l'impact des modifications proposées avant de fusionner la demande d'extraction. Pour de plus amples informations, veuillez consulter [Permet CloudFormation de publier un résumé des modifications de la pile dans les pull requests](#).

Fichier de déploiement de pile

Un fichier de déploiement de pile est un fichier au format standard JSON (JavaScript Object Notation) ou YAML qui contient les paramètres et les valeurs qui gèrent votre CloudFormation pile. Il est surveillé pour détecter les modifications. Lorsque les modifications apportées au fichier sont validées dans le référentiel, la pile associée est automatiquement mise à jour.

Le fichier de déploiement de pile contient une paire clé-valeur et deux dictionnaires :

- `template-file-path`

Il s'agit du chemin complet du référentiel pour le fichier CloudFormation modèle. Le fichier modèle déclare les ressources pour la CloudFormation pile associée à ce fichier de déploiement.

- `parameters`

Le dictionnaire de paramètres contient des paires clé-valeur qui configurent les ressources de la pile. Un fichier de déploiement de pile peut comporter jusqu'à 50 paramètres.

- `tags`

Le dictionnaire des balises contient des paires clé-valeur facultatives que vous pouvez utiliser pour identifier et classer les ressources de la pile. Un fichier de déploiement de pile peut comporter jusqu'à 50 balises.

Vous pouvez fournir votre propre fichier de déploiement de pile, ou demander à la synchronisation Git d'en créer un pour vous et d'envoyer automatiquement une demande d'extraction à votre référentiel.

Vous pouvez gérer les paramètres et les balises en modifiant le fichier de déploiement de pile et en validant les modifications dans le référentiel.

Voici un exemple de fichier de déploiement de pile avec synchronisation Git :

```
template-file-path: fargate-srvc/my-stack-template.yaml

parameters:
  image: public.ecr.aws/lts/nginx:latest
  task_size: x-small
  max_capacity: 5
  port: 8080
  env: production
tags:
  cost-center: '123456'
  org: 'AWS'
```

CloudFormation fichier modèle

Un fichier modèle contient une déclaration des AWS ressources qui constituent une CloudFormation pile. Avec la synchronisation Git, le fichier modèle est stocké dans votre référentiel Git et référencé par le fichier de déploiement de la pile. Vous pouvez gérer la pile en modifiant le fichier modèle et en validant les modifications dans le référentiel.

Pour de plus amples informations, veuillez consulter [Utilisation de CloudFormation modèles](#).

Référentiel de définition du modèle

Le référentiel de définition du modèle est le référentiel Git auquel est lié CloudFormation via Git sync. Le référentiel est surveillé pour détecter les modifications apportées au CloudFormation modèle et au fichier de déploiement de la pile. Lorsque vous validez les modifications apportées au fichier, la pile associée est automatiquement mise à jour.

Important

Lorsque vous configurez le référentiel de définition du modèle dans la console de synchronisation Git, sélectionnez le référentiel et la branche corrects à partir de la connexion Git. Git sync surveille uniquement le référentiel et la branche configurés pour détecter les modifications apportées au CloudFormation modèle et au fichier de déploiement de la pile.

Conditions préalables à la synchronisation des piles vers un référentiel Git à l'aide de la synchronisation Git

Avant de synchroniser une CloudFormation pile avec votre dépôt Git, vérifiez que les conditions suivantes sont remplies.

Rubriques

- [Référentiel Git](#)
- [CloudFormation modèle](#)
- [Rôle de service de synchronisation Git](#)
- [Autorisations IAM pour les utilisateurs de la console](#)

Référentiel Git

Vous devez disposer d'un référentiel Git hébergé sur l'une des plateformes suivantes.

- [GitHub](#)
- [GitHub Entreprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLabautogéré](#)

Le référentiel peut être public ou privé. Vous devrez connecter ce dépôt Git via CloudFormation la [console Connections](#).

CloudFormation modèle

Votre dépôt Git doit contenir un [fichier CloudFormation modèle](#) enregistré dans la branche à laquelle vous souhaitez vous connecter avec Git sync. Ce modèle sera référencé par le [fichier de déploiement de la pile](#).

Rôle de service de synchronisation Git

La synchronisation Git nécessite un rôle IAM. Vous pouvez choisir de créer un rôle IAM pour votre pile lorsque vous configurez la synchronisation Git, ou vous pouvez utiliser un rôle existant.

 Note

Un rôle IAM généré automatiquement applique des autorisations uniquement à la pile pour laquelle le rôle est généré. Pour réutiliser un rôle IAM généré automatiquement, vous devez modifier le rôle pour la nouvelle pile.

Autorisations requises pour le rôle de service de synchronisation Git

Le rôle IAM que vous attribuez à la synchronisation Git nécessite les autorisations suivantes.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:ListChangeSets`
- `cloudformation:ValidateTemplate`
- `events:PutRule`
- `events:PutTargets`

 Note

Les autorisations requises ci-dessous sont automatiquement ajoutées aux rôles IAM générés par la synchronisation Git.

L'exemple de rôle IAM suivant inclut les autorisations prérequis pour la synchronisation Git.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {
```

```

        "Sid": "SyncToCloudFormation",
        "Effect": "Allow",
        "Action": [
            "cloudformation:CreateChangeSet",
            "cloudformation:DeleteChangeSet",
            "cloudformation:DescribeChangeSet",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStacks",
            "cloudformation:ExecuteChangeSet",
            "cloudformation:GetTemplate",
            "cloudformation:ListChangeSets",
            "cloudformation:ListStacks",
            "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
    },
    {
        "Sid": "PolicyForManagedRules",
        "Effect": "Allow",
        "Action": [
            "events:PutRule",
            "events:PutTargets"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "events:ManagedBy":
["cloudformation.sync.codeconnections.amazonaws.com"]
            }
        }
    },
    {
        "Sid": "PolicyForDescribingRule",
        "Effect": "Allow",
        "Action": "events:DescribeRule",
        "Resource": "*"
    }
]
}

```

Politique d'approbation

Vous devez fournir la politique d'approbation suivante lorsque vous créez le rôle pour définir la relation d'approbation.

JSON

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

Nous vous recommandons d'utiliser les clés de condition `aws:SourceArn` et `aws:SourceAccount` pour vous protéger contre le problème de l'adjoint confus. Le compte source est votre identifiant de compte et l'ARN source est l'ARN de la connexion au [CodeConnections](#) service qui permet de se connecter CloudFormation à votre dépôt Git.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {

```

```
    "aws:SourceAccount": "123456789012"
  },
  "ArnLike": {
    "aws:SourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/EXAMPLE64-8aad-4d5d-8878-dfcab0bc441f"
  }
}
]
```

Pour de plus amples informations sur le problème de l'adjoint confus, veuillez consulter [Prévention du problème de l'adjoint confus entre services](#).

Autorisations IAM pour les utilisateurs de la console

Pour configurer correctement Git Sync via la CloudFormation console, les utilisateurs finaux doivent également disposer d'autorisations via IAM.

Les autorisations codeconnections suivantes sont requises pour créer et gérer la connexion à votre référentiel Git.

- codeconnections:CreateRepositoryLink
- codeconnections:CreateSyncConfiguration
- codeconnections>DeleteRepositoryLink
- codeconnections>DeleteSyncConfiguration
- codeconnections:GetRepositoryLink
- codeconnections:GetSyncConfiguration
- codeconnections:ListRepositoryLinks
- codeconnections:ListSyncConfigurations
- codeconnections:ListTagsForResource
- codeconnections:TagResource
- codeconnections:UntagResource
- codeconnections:UpdateRepositoryLink
- codeconnections:UpdateSyncBlocker
- codeconnections:UpdateSyncConfiguration

- `codeconnections:UseConnection`

Les utilisateurs de console doivent également disposer des autorisations `cloudformation` suivantes pour afficher et gérer les piles pendant le processus de configuration de la synchronisation Git.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:GetTemplate`
- `cloudformation:ListChangeSets`
- `cloudformation:ListStacks`
- `cloudformation:ValidateTemplate`

Note

Bien que les autorisations de modification (`cloudformation:CreateChangeSet`, `cloudformation>DeleteChangeSet`, `cloudformation:DescribeChangeSet`, `cloudformation:ExecuteChangeSet`) ne soient pas strictement nécessaires pour une utilisation exclusive de la console, elles sont recommandées pour permettre l'inspection complète de la pile et les capacités de gestion.

L'exemple de politique IAM suivant inclut les autorisations utilisateur nécessaires pour configurer la synchronisation Git via la console.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Sid": "CodeConnectionsPermissions",
        "Effect": "Allow",
        "Action": [
            "codeconnections:CreateRepositoryLink",
            "codeconnections:CreateSyncConfiguration",
            "codeconnections>DeleteRepositoryLink",
            "codeconnections>DeleteSyncConfiguration",
            "codeconnections:GetRepositoryLink",
            "codeconnections:GetSyncConfiguration",
            "codeconnections:ListRepositoryLinks",
            "codeconnections:ListSyncConfigurations",
            "codeconnections:ListTagsForResource",
            "codeconnections:TagResource",
            "codeconnections:UntagResource",
            "codeconnections:UpdateRepositoryLink",
            "codeconnections:UpdateSyncBlocker",
            "codeconnections:UpdateSyncConfiguration",
            "codeconnections:UseConnection",
            "codeconnections:CreateForcedTargetSync",
            "codeconnections:CreatePullRequestForResource"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudFormationConsolePermissions",
        "Effect": "Allow",
        "Action": [
            "cloudformation:CreateChangeSet",
            "cloudformation>DeleteChangeSet",
            "cloudformation:DescribeChangeSet",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStacks",
            "cloudformation:ExecuteChangeSet",
            "cloudformation:GetTemplate",
            "cloudformation:ListChangeSets",
            "cloudformation:ListStacks",
            "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
    }
]
}

```


 Note

Lorsque vous créez une politique IAM qui inclut les autorisations `codeconnections:CreateForcedTargetSync` et `codeconnections:CreatePullRequestForResource`, un avertissement peut s'afficher dans la console IAM indiquant que ces actions n'existent pas. Cet avertissement peut être ignoré et la politique sera toujours créée avec succès. Ces autorisations sont requises pour certaines opérations de synchronisation Git même si elles ne sont pas reconnues par la console IAM.

Créez une pile à partir du code source du référentiel avec la synchronisation Git

Cette rubrique explique comment créer une CloudFormation pile synchronisée avec un dépôt Git avec Git sync.

 Important

Avant de continuer, remplissez l'ensemble des [conditions préalables](#) de la section précédente.

Créer une pile à partir du code source du référentiel

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le Région AWS pour créer le stack in.
3. Sur la page Piles, choisissez Créer une pile, puis Avec de nouvelles ressources (standard).
4. Sur la page Créer une pile, procédez de la manière suivante :
 - a. Pour la Condition préalable – Préparer le modèle, laissez la case Choisir un modèle existant cochée.
 - b. Pour Spécifier le modèle, choisissez Sync. depuis Git, puis Suivant.

5. Sur la page Spécifier les détails de la pile, entrez un nom pour la pile dans la zone Nom de la pile. Les noms de pile peuvent contenir des lettres (A à Z et a à z), des chiffres (0 à 9) et des tirets (-).
6. Pour le fichier de déploiement Stack, création du fichier de déploiement :
 - Si vous n'avez pas créé de fichier de déploiement de pile et ne l'avez pas ajouté à votre référentiel, sélectionnez Créer le fichier à l'aide des paramètres suivants et le placer dans mon référentiel.
 - Si votre référentiel contient un fichier de déploiement de pile, choisissez Je fournis mon propre fichier dans mon référentiel.
7. Pour le référentiel de définition de modèles, choisissez Choisir un dépôt Git lié pour choisir un dépôt Git auquel il est déjà lié CloudFormation, ou Lier un dépôt Git pour en lier un nouveau. Si vous choisissez Lier un référentiel Git, procédez comme suit :
 - a. Pour Sélectionner un fournisseur de référentiel, choisissez l'une des options suivantes :
 - GitHub
 - GitHub Serveur d'entreprise
 - GitLab
 - Bitbucket
 - GitLab autogéré
 - b. Pour la connexion, choisissez une connexion dans la liste. Si aucune option n'apparaît dans la liste Connexion, choisissez Ajouter une nouvelle connexion pour accéder à la [console Connexions](#) et créer une connexion à votre référentiel.
8. Dans la liste Référentiel, sélectionnez le référentiel Git qui contient votre fichier modèle de pile.
9. Dans la liste Branche, sélectionnez la branche que vous souhaitez que la synchronisation Git surveille.

 Note

Git sync surveille uniquement les modifications apportées au CloudFormation modèle et aux fichiers de déploiement de la pile dans la branche sélectionnée. Toutes les modifications que vous souhaitez appliquer à votre pile doivent être validées dans cette branche.

10. Pour le chemin du fichier de déploiement, spécifiez le chemin complet, y compris le nom du fichier de déploiement de la pile, à partir de la racine de la branche de votre référentiel.

S' CloudFormation il génère le fichier pour vous, c'est ici que le fichier sera enregistré dans votre dépôt. Si vous fournissez le fichier, il s'agit de l'emplacement du fichier dans votre référentiel.

11. Ajoutez un rôle IAM. Le rôle IAM inclut les autorisations requises CloudFormation pour synchroniser la pile depuis votre référentiel Git. Vous pouvez choisir Nouveau rôle IAM pour générer un nouveau rôle, ou choisir Rôle IAM existant pour sélectionner un rôle existant dans votre Compte AWS. Si vous choisissez de générer un nouveau rôle, les autorisations requises sont incluses dans le rôle.
12. Activez ou désactivez les commentaires en cas de demande d'extraction :
 - Pour que les informations relatives à la CloudFormation post-modification soient définies dans les pull requests pour les mises à jour de la pile, maintenez le bouton Activer le commentaire sur la pull request activé.
 - Si vous désactivez cette option, je ne CloudFormation décrirai pas les différences entre la configuration actuelle de la pile et les modifications proposées dans les pull requests lorsque les fichiers de dépôt sont mis à jour.
13. Pour le chemin du fichier modèle, spécifiez le chemin complet de la racine de votre référentiel pour le fichier modèle de la pile.
14. (Facultatif) Pour spécifier les paramètres de la pile, sélectionnez Ajouter un paramètre, indiquez une clé et une valeur pour chaque paramètre, puis sélectionnez Suivant. Pour de plus amples informations, veuillez consulter [Fichier de déploiement de pile](#).

Par exemple, pour spécifier un paramètre **port=8080** dans votre fichier de déploiement de pile, procédez comme suit :

- a. Choisissez Ajouter un paramètre.
 - b. Pour Clé, entrez **port**.
 - c. Pour le champ Value (Valeur), entrez **8080**.
15. (Facultatif) Pour spécifier des balises de pile, sélectionnez Ajouter une nouvelle balise, fournissez une clé et une valeur pour chaque balise, puis sélectionnez Suivant. Pour de plus amples informations, veuillez consulter [Fichier de déploiement de pile](#).
 16. Choisissez Suivant pour passer à Configurer les options de la pile. Pour plus d'informations sur les options de configuration de la pile, consultez [Configuration des options de pile](#).

Une fois la configuration de votre pile terminée, sélectionnez Suivant pour continuer.

17. Vérifiez les paramètres de votre pile et confirmez les points suivants :

- Le modèle de pile est correctement configuré et défini sur Synchronisation à partir de Git.
- Le fichier de déploiement est correctement configuré.
- Le référentiel de définition du modèle est correctement configuré, en particulier en ce qui concerne la sélection du Référentiel et du Nom de branche corrects.
- L'aperçu du fichier de déploiement est correct et contient les paramètres et valeurs attendus.

18. Choisissez Envoyer pour créer la pile.

Après avoir sélectionné Soumettre, une demande d'extraction est automatiquement créée dans votre référentiel Git. Vous devez fusionner cette demande d'extraction dans votre référentiel Git pour créer votre pile. Une fois la pile créée, CloudFormation surveille les modifications apportées à votre dépôt Git.

Mettez à jour votre pile depuis votre référentiel Git

Pour mettre à jour la pile, apportez les modifications directement dans votre fichier modèle ou dans le fichier de déploiement de la pile dans votre référentiel Git. Une fois que vous avez validé vos modifications dans la branche surveillée, la pile est CloudFormation automatiquement mise à jour. Si vous utilisez des requêtes pull, une requête pull est automatiquement créée dans votre référentiel Git avant la mise à jour de la pile. Vous devez fusionner cette demande d'extraction dans votre référentiel Git pour mettre à jour votre pile.

Dans la CloudFormation console, vous pouvez sélectionner la pile et choisir l'onglet Git sync pour afficher les informations relatives à l'état de la pile et aux événements de synchronisation. Pour de plus amples informations, veuillez consulter [Tableau de bord d'état de la synchronisation Git](#).

Permet CloudFormation de publier un résumé des modifications de la pile dans les pull requests

Cette rubrique explique comment activer la publication d'un résumé des modifications apportées CloudFormation à la pile dans les pull requests dans votre dépôt Git.

En activant la fonctionnalité de commentaires sur les pull requests, vous CloudFormation autorisez la publication d'un commentaire décrivant les différences entre la configuration de la pile actuelle et

les modifications proposées lors de la mise à jour des fichiers de dépôt. Ce commentaire fournit un résumé des ressources qui seront ajoutées, modifiées ou supprimées, vous permettant d'effectuer une révision de code complète avant de fusionner la requête pull.

Activation des commentaires sur les requêtes pull pour une nouvelle pile (console)

Lorsque vous créez la pile, sur la page Spécifier les détails de la pile, sous Référentiel de définitions de modèles, assurez-vous que le bouton Activer les commentaires sur les requêtes pull est activé. Il s'agit du paramètre par défaut pour les nouvelles piles.

Activation des commentaires sur les requêtes pull pour une pile existante (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le modèle dans Région AWS le quel vous avez créé votre stack.
3. Sur la page Piles, choisissez la pile en cours d'exécution que vous souhaitez mettre à jour.
4. Choisissez l'onglet Synchronisation depuis Git, puis Modifier.
5. Sur la page Modifier les paramètres de synchronisation depuis Git, sous Référentiel de définitions de modèles, activez l'option Activer les commentaires sur les requêtes pull.
6. Choisissez Mettre à jour la configuration.

Tableau de bord d'état de la synchronisation Git

Pour afficher l'état d'un déploiement de synchronisation AWS CloudFormation Git, sélectionnez la pile dans la CloudFormation console et choisissez l'onglet Git sync.

L'onglet Synchronisation Git est divisé en deux volets, État de la synchronisation Git et Derniers événements de synchronisation.

État de la synchronisation Git

Le volet supérieur fournit les informations suivantes sur la configuration de la synchronisation Git de la pile.

Référentiel

Lien vers le référentiel connecté à la synchronisation Git

Fournisseur du référentiel

Nom du fournisseur du référentiel

Branche

Nom de la branche surveillée par la synchronisation Git

Chemin du fichier de déploiement

Chemin complet vers le fichier de déploiement de cette pile

État de la synchronisation du référentiel

État de la dernière opération de synchronisation

Message d'état de la synchronisation du référentiel

Message de la dernière opération de synchronisation

État de la synchronisation Git

État de la synchronisation Git de cette pile

État du provisionnement

État de l'opération de provisionnement

Dans la partie supérieure droite du volet, utilisez les boutons suivants pour modifier ou mettre à jour la synchronisation Git :

- **Modifier** : modifiez la configuration de la synchronisation Git.
- **Réessayer la dernière validation** : mettez à jour la pile en fonction de la dernière validation dans le référentiel.
- **Déconnecter** : déconnectez la synchronisation Git de la pile.
- **Actualiser** : actualisez le volet d'état de la synchronisation Git.

Derniers événements de synchronisation

Le volet Derniers événements de synchronisation affiche une table des validations appliquées à la pile.

Vous pouvez trier la table à l'aide des flèches situées dans l'en-tête de chaque colonne. La table peut être triée par ordre croissant ou décroissant selon les critères suivants :

- Date
- ID de validation
- Event
- Date
- Type d'événement

États de pile pris en charge

La synchronisation Git ne peut être configurée pour une pile que si celle-ci se trouve dans l'un des états pris en charge suivants :

- CREATE_COMPLETE
- UPDATE_COMPLETE
- UPDATE_ROLLBACK_COMPLETE
- IMPORT_COMPLETE
- IMPORT_ROLLBACK_COMPLETE

Le tableau suivant répertorie l'ensemble des codes d'état de la pile avec leurs descriptions :

État de la pile et statut détaillé facultatif	Description
CREATE_COMPLETE	Réussite de la création d'une ou de plusieurs piles.
CREATE_IN_PROGRESS	Création en cours d'une ou de plusieurs piles.
CREATE_FAILED	Échec de la création d'une ou de plusieurs piles. Affichez les événements de la pile pour voir les messages d'erreur associés. Les raisons possibles d'un échec de création incluent des autorisations insuffisantes pour utiliser toutes les ressources de la pile, des valeurs de paramètres rejetées par un AWS service ou un délai d'attente lors de la création des ressources.

État de la pile et statut détaillé facultatif	Description
DELETE_COMPLETE	Réussite de la suppression d'une ou de plusieurs piles. Les piles supprimées sont conservées et consultables pendant 90 jours.
DELETE_FAILED	Échec de la suppression d'une ou de plusieurs piles. Comme la suppression a échoué, certaines ressources sont peut-être encore en cours d'exécution. Cependant, vous ne pouvez pas utiliser ni mettre à jour la pile. Supprimez la pile à nouveau ou consultez les événements correspondants pour voir les messages d'erreur associés.
DELETE_IN_PROGRESS	Suppression en cours d'une ou de plusieurs piles.
REVIEW_IN_PROGRESS	Création continue d'une ou de plusieurs piles avec un StackId attendu, mais sans aucun modèle ni ressource.  Important Une pile associée à ce code de statut est comptabilisée dans le nombre maximal possible de piles.
ROLLBACK_COMPLETE	Réussite de la suppression d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile. La pile revient à son état de fonctionnement précédent. Toutes les ressources qui ont été créées au cours de l'action de création de la pile sont supprimées. Ce statut existe uniquement après un échec de création de la pile. Il signifie que toutes les opérations de la pile partiellement créée ont été correctement nettoyées. Dans ce cas-là, seule une opération de suppression peut être effectuée.

État de la pile et statut détaillé facultatif	Description
ROLLBACK_FAILED	Échec de la suppression d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile. Supprimez la pile ou consultez les événements correspondants pour voir les messages d'erreur associés.
ROLLBACK_IN_PROGRESS	Suppression en cours d'une ou de plusieurs piles après un échec de création ou après l'annulation explicite de la création de la pile.
UPDATE_COMPLETE	Réussite de la mise à jour d'une ou de plusieurs piles.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Suppression en cours d'anciennes ressources pour une ou plusieurs piles après une mise à jour réussie. Pour les mises à jour de pile qui nécessitent le remplacement de ressources, CloudFormation crée d'abord les nouvelles ressources, puis supprime les anciennes ressources afin de réduire les interruptions de votre pile. Dans cet état, la pile a été mise à jour et est utilisable, mais les anciennes ressources CloudFormation sont toujours supprimées.
UPDATE_FAILED	Réussite de la mise à jour d'une ou de plusieurs piles. Affichez les événements de la pile pour voir les messages d'erreur associés.
UPDATE_IN_PROGRESS	Mise à jour en cours d'une ou de plusieurs piles.
UPDATE_ROLLBACK_COMPLETE	Réussite de la restauration de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Suppression en cours des nouvelles ressources d'une ou de plusieurs piles après un échec de mise à jour. Dans cet état, la pile a été rétablie à son état de fonctionnement précédent et est utilisable, mais CloudFormation les nouvelles ressources créées lors de la mise à jour de la pile sont toujours supprimées.

État de la pile et statut détaillé facultatif	Description
UPDATE_ROLLBACK_FAILED	Échec de la restauration de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour. Dans ce cas là, vous pouvez supprimer la pile ou poursuivre la restauration . Vous devrez peut-être corriger les erreurs avant que votre pile puisse revenir à un état de fonctionnement. Vous pouvez également contacter Support pour restaurer la pile dans un état utilisable.
UPDATE_ROLLBACK_IN_PROGRESS	Restauration en cours de l'état de fonctionnement d'une ou plusieurs piles après un échec de mise à jour.
IMPORT_IN_PROGRESS	L'opération d'importation est en cours.
IMPORT_COMPLETE	L'opération d'importation a été exécutée avec succès pour toutes les ressources de la pile qui prennent en charge <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	L'importation sera restaurée à la configuration précédente du modèle.
IMPORT_ROLLBACK_FAILED	L'opération de restauration de l'importation a échoué pour au moins une ressource de la pile. Les résultats seront disponibles pour les ressources importées CloudFormation avec succès.
IMPORT_ROLLBACK_COMPLETE	L'importation a été restaurée avec succès à la configuration du modèle précédente.

Gestion des extensions avec le CloudFormation registre

Le CloudFormation registre sert de hub centralisé pour gérer les extensions qui peuvent être intégrées dans les CloudFormation modèles de votre Compte AWS. Les extensions incluent les types de ressources, les modules et les Hooks fournis AWS par des éditeurs tiers ou tiers, ainsi que vos propres extensions personnalisées. Le registre facilite la découverte et le provisionnement des extensions dans vos CloudFormation modèles de la même manière que vous utilisez les ressources AWS fournies.

Cette section décrit comment utiliser le CloudFormation registre pour activer des extensions tierces sur votre compte, notamment :

- Activation d'extensions publiques
- Enregistrement et activation d'extensions privées

Rubriques

- [Documents associés](#)
- [CloudFormation concepts de registre](#)
- [Afficher les extensions disponibles et activées dans le CloudFormation registre](#)
- [Utiliser des extensions publiques tierces depuis le CloudFormation registre](#)
- [Utilisez des extensions privées tierces qui ont été partagées avec vous](#)
- [Modifier les données de configuration des extensions de votre compte](#)
- [Enregistrez les types de ressources dans AWS Config](#)

Documents associés

Si vous êtes un développeur intéressé par la création de vos propres extensions, consultez la documentation suivante :

- [Développement de modules à l'aide de la CloudFormation CLI](#) dans le guide de l'utilisateur de l'interface de ligne de CloudFormation commande
- [Création de types de ressources à l'aide de la CloudFormation CLI](#) dans le guide de l'utilisateur de l'interface de ligne de CloudFormation commande

- [Développement de Hooks personnalisés à l'aide de la CloudFormation CLI](#) dans le guide de l'utilisateur CloudFormation des Hooks

En outre, tous les types de AWS ressources provisionnables disponibles dans le CloudFormation registre peuvent être utilisés avec le API de commande du Cloud AWS, leurs attributs et propriétés étant définis dans un schéma JSON standard. Pour plus d'informations, consultez le [Guide de l'utilisateur de l'API Cloud Control](#). Lorsque vous utilisez l'API Cloud Control pour effectuer des opérations CRUDL (Create, Read, Update, Delete, List) sur AWS des ressources, vous ne pouvez le faire que sur AWS des ressources qui vous appartiennent Compte AWS.

CloudFormation concepts de registre

Cette rubrique explique les concepts clés qui vous aideront à comprendre le CloudFormation registre et à commencer à l'utiliser.

Types d'extension

Le CloudFormation registre propose les types d'extensions suivants :

Crochets

Les hooks sont des contrôles de validation qui inspectent vos piles ou des ressources spécifiques avant de les CloudFormation créer, de les mettre à jour ou de les supprimer. Les hooks peuvent également être invoqués lors des opérations de création d'ensembles de modifications. Ils fournissent un mécanisme permettant de faire respecter les normes organisationnelles et les meilleures pratiques en validant les configurations des ressources par rapport à des exigences spécifiques. Si un Hook détecte des configurations non conformes à sa logique, il peut soit émettre un avertissement, soit interrompre le processus de provisionnement afin d'empêcher le déploiement de ressources non conformes. Pour plus d'informations, consultez le [Guide de l'utilisateur des hooks CloudFormation](#).

Pour obtenir de la documentation sur la configuration des Hooks à l'aide de la CloudFormation console, consultez les sections suivantes du guide de l'utilisateur des CloudFormation Hooks :

- [AWS Control Tower contrôles proactifs sous forme de Hooks](#)
- [Hooks Guard](#)
- [Hooks Lambda](#)

Modules

Les modules sont des configurations de ressources réutilisables qui peuvent être incluses dans plusieurs modèles de CloudFormation pile. Ils simplifient la création et la maintenance des CloudFormation modèles en encapsulant des configurations de ressources complexes ou fréquemment utilisées dans des composants réutilisables. Cela favorise la cohérence et la normalisation dans l'ensemble des déploiements d'infrastructure de votre organisation.

Resource types

Les types de ressources vous permettent de modéliser et d'automatiser des ressources tierces ou des ressources personnalisées qui ne sont pas prises en charge de manière native par CloudFormation. En développant des types de ressources, vous pouvez étendre les capacités CloudFormation de provisionnement et de gestion des ressources provenant de divers services tiers.

Extensions publiques

Les extensions publiques sont des CloudFormation extensions publiées publiquement dans le registre pour être utilisées par tous Comptes AWS. Il s'agit des licences suivantes :

- **AWS extensions** : les extensions publiées par AWS sont toujours publiques et activées par défaut. Vous n'avez donc aucune action à effectuer avant de les utiliser dans votre compte. AWS contrôle la gestion des versions de ces extensions, afin que vous utilisiez toujours la dernière version disponible.
- **Extensions tierces** : ces extensions sont mises à disposition pour une utilisation générale par des éditeurs autres que AWS. Pour utiliser une extension publique tierce, vous devez d'abord l'activer dans votre compte et votre région.

Extensions activées

Dans votre CloudFormation registre, vous Compte AWS trouverez trois types d'extensions activées :

- **AWS extensions** — Toutes les extensions AWS publiques sont automatiquement activées.
- **Tiers activés** : il s'agit de copies locales d'extensions publiques tierces que vous avez explicitement activées pour votre compte et votre région. Lorsque vous activez une extension publique tierce, elle CloudFormation crée une copie locale dans le registre de votre compte. Pour de plus amples

informations, veuillez consulter [Utiliser des extensions publiques tierces depuis le CloudFormation registre](#).

- Enregistrées en privé — Il s'agit d'extensions privées qui ne figurent pas dans le CloudFormation registre public. Il peut s'agir d'extensions que vous avez créées vous-même ou qui vous ont été fournies par votre organisation ou d'autres tiers. Pour utiliser une extension privée dans votre compte, vous devez d'abord l'enregistrer. L'enregistrement de l'extension télécharge une copie dans le CloudFormation registre de votre compte et l'active. Pour de plus amples informations, veuillez consulter [Utilisez des extensions privées tierces qui ont été partagées avec vous](#).

Utiliser des extensions enregistrées à titre privé et des extensions publiques activées provenant d'éditeurs tiers dans votre compte revient à les utiliser dans un environnement de test (sandbox). Les extensions sont gérées avec un contrôle de version, leur comportement de provisionnement est donc lié à une version spécifique. Par conséquent, ces extensions fonctionnent de la même manière que les extensions publiques, en respectant les mêmes règles spécifiques à chaque version.

Note

Les extensions enregistrées de manière privée et les extensions publiques tierces activées peuvent implémenter des gestionnaires d'événements qui s'exécutent lors des opérations de création, lecture, mise à jour, listage et suppression. L'utilisation de ces extensions dans vos CloudFormation piles peut entraîner des frais sur votre compte, en plus des frais liés aux ressources créées. Pour en savoir plus, consultez [Pricing CloudFormation](#) (Tarification).

Afficher les extensions disponibles et activées dans le CloudFormation registre

Pour consulter les extensions disponibles et activées dans le CloudFormation registre, vous pouvez utiliser le AWS Management Console ou le AWS CLI.

Pour afficher les extensions disponibles et activées (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.

3. Dans le volet de navigation, sous Registre, choisissez la catégorie d'extension que vous souhaitez afficher :
 - Public extensions (Extensions publiques) affiche les extensions publiques disponibles dans votre compte.
 1. Dans Filtre, Type d'extension, choisissez votre type d'extension : types de ressources, modules ou Hooks.
 2. Pour Filter, diffuseur de publication, choisissez un diffuseur de publication : AWS ou un tiers.
 - Activated extensions (Extensions activées) affiche les extensions publiques et privées qui ont été activées dans votre compte.
 1. Choisissez votre type d'extension : Resource types (Types de ressource), Modules ou Hooks.
 2. Utilisez le menu déroulant Filtre pour sélectionner davantage les extensions à afficher :
 - AWS— répertorie les extensions publiées par AWS. Les extensions publiées par AWS sont activées par défaut.
 - Tierce : répertorie toutes les extensions publiques des éditeurs autres AWS que celles que vous avez activées dans ce compte.
 - Registered (Enregistré) : répertorie toutes les extensions privées que vous avez activées sur ce compte.
 - Publisher (Éditeur) affiche toutes les extensions publiques que vous avez publiées à l'aide de ce compte.
4. Recherchez ou sélectionnez le nom de l'extension pour afficher les détails de l'extension.

Pour afficher les extensions disponibles et activées (AWS CLI)

Utilisez la commande [list-types](#).

Utiliser des extensions publiques tierces depuis le CloudFormation registre

Pour utiliser une extension publique tierce dans votre modèle, vous devez d'abord activer l'extension pour le compte et la région où vous souhaitez l'utiliser. L'activation d'une extension permet de l'utiliser dans les opérations de pile du compte et de la région où elle est activée.

Lorsque vous activez une extension publique tierce, CloudFormation crée une entrée dans le registre des extensions de votre compte pour l'extension activée en tant qu'extension privée. Cela vous permet de définir les propriétés de configuration incluses dans l'extension. Les propriétés de configuration définissent la manière dont l'extension est configurée pour une région donnée Compte AWS .

Outre la définition des propriétés de configuration, vous pouvez également personnaliser l'extension de la manière suivante :

- Spécifiez le rôle d'exécution CloudFormation utilisé pour activer l'extension, en plus de configurer la journalisation de l'extension.
- Spécifiez si l'extension est automatiquement mise à jour lorsqu'une nouvelle version mineure ou corrective est disponible.
- Spécifiez un alias à utiliser à la place du nom de l'extension publique tierce. Cela permet d'éviter les collisions de noms entre les extensions tierces.

Rubriques

- [Configurer un rôle d'exécution avec des autorisations IAM et une politique de confiance pour l'accès public aux extensions](#)
- [Utiliser automatiquement les nouvelles versions des extensions](#)
- [Utilisez des alias pour faire référence aux extensions](#)
- [AWS CLI Commandes couramment utilisées pour travailler avec des extensions publiques](#)
- [Activation d'une extension publique tierce dans votre compte](#)
- [Mettre à jour une extension tierce publique dans votre compte](#)
- [Désactivation des extensions publiques tierces dans votre compte](#)

Configurer un rôle d'exécution avec des autorisations IAM et une politique de confiance pour l'accès public aux extensions

Lorsque vous activez une extension publique depuis le CloudFormation registre, vous pouvez fournir un rôle d'exécution qui donne CloudFormation les autorisations nécessaires pour invoquer cette extension dans votre région Compte AWS et dans votre région.

Les autorisations requises pour le rôle d'exécution sont définies dans la section handler du schéma d'extension. Vous devez créer une politique IAM qui accorde les autorisations spécifiques requises par l'extension et l'associer au rôle d'exécution.

Outre la politique d'autorisations, le rôle d'exécution doit également disposer d'une politique de confiance qui permet CloudFormation d'assumer le rôle. Suivez les instructions de la section [Créer un rôle à l'aide de stratégies de confiance personnalisées](#) dans le Guide de l'utilisateur IAM pour créer un rôle avec une stratégie de confiance personnalisée.

Relation d'approbation

Vous trouverez ci-dessous des exemples de politiques de confiance que vous pouvez utiliser.

Vous pouvez éventuellement restreindre la portée de l'autorisation pour la prévention des délaisés croisés entre services en utilisant une ou plusieurs clés de contexte de condition globales avec le champ `Condition`. Pour de plus amples informations, veuillez consulter [Prévention du problème de l'adjoint confus entre services](#).

- Définissez la valeur `aws:SourceAccount` sur l'ID de votre compte.
- Définissez la valeur `aws:SourceArn` sur l'ARN de votre extension.

Exemple de politique d'approbation 1

Voici un exemple de stratégie de confiance des rôles IAM pour une extension de type de ressource.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resources.cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```

    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Organization-Service-Resource"
    }
  }
}
]
}

```

Exemple de politique d'approbation 2

Voici un exemple de stratégie de confiance des rôles IAM pour une extension Hook.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com",
          "hooks.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/hook/Organization-Service-Hook"
        }
      }
    }
  ]
}

```

Utiliser automatiquement les nouvelles versions des extensions

Lorsque vous activez une extension, vous pouvez également spécifier le type d'extension pour utiliser la dernière version mineure. Votre type d'extension met à jour la version mineure chaque fois que l'éditeur publie une nouvelle version de votre extension activée.

Par exemple, la prochaine fois que vous effectuez une opération de pile, telle que la création ou la mise à jour d'une pile, à l'aide d'un modèle incluant cette extension, vous CloudFormation utiliserez la nouvelle version mineure.

La mise à jour vers une nouvelle version d'extension, soit automatiquement soit manuellement, n'affecte pas les instances d'extension déjà allouées dans les piles.

CloudFormation considère les mises à jour des versions majeures des extensions comme pouvant contenir des modifications majeures, et vous oblige donc à effectuer une mise à jour manuelle vers une nouvelle version majeure d'une extension.

Les extensions publiées par AWS sont activées par défaut pour tous les comptes et régions où elles sont disponibles, et utilisent toujours la dernière version disponible pour chacun d'entre eux Région AWS.

Important

Étant donné que vous contrôlez si et quand les extensions sont mises à jour vers la dernière version dans votre compte, vous pourriez vous retrouver avec différentes versions de la même extension déployée dans différents comptes et régions.

Cela peut entraîner des résultats inattendus lors de l'utilisation du même modèle contenant cette extension, pour ces comptes et régions.

Utilisez des alias pour faire référence aux extensions

Vous ne pouvez pas activer plus d'une extension avec un nom donné dans un Compte AWS et une région donnés. Étant donné que différents éditeurs peuvent proposer des extensions publiques portant le même nom d'extension CloudFormation, vous pouvez spécifier un alias pour toute extension publique tierce que vous activez.

Si vous spécifiez un alias pour l'extension, CloudFormation traite-le comme le nom du type d'extension dans le compte et la région. Vous devez utiliser l'alias pour faire référence à l'extension dans vos modèles, vos appels d'API et votre CloudFormation console.

Des alias d'extension doivent être uniques au sein d'un compte et d'une région donnés. Vous pouvez activer la même ressource publique plusieurs fois dans le même compte et la même région, en utilisant différents alias de nom de type.

⚠ Important

Bien que les alias d'extension ne doivent être uniques que dans un compte et une région donnés, nous recommandons vivement aux utilisateurs de ne pas attribuer le même alias à différentes extensions publiques tierces sur l'ensemble des comptes et des régions. Cela pourrait entraîner des résultats inattendus lors de l'utilisation d'un modèle contenant l'alias d'extension pour plusieurs comptes ou régions.

AWS CLI Commandes couramment utilisées pour travailler avec des extensions publiques

Les commandes couramment utilisées pour travailler avec les extensions publiques sont les suivantes :

- [activate-type](#) pour activer un module ou un type de ressource tiers public dans votre compte.
- [set-type-configuration](#) pour spécifier les données de configuration d'une extension dans votre compte et pour désactiver et activer les Hooks.
- [list-types](#) pour répertorier les extensions de votre compte.
- [describe-type](#) pour renvoyer des informations détaillées sur une extension ou une version d'extension spécifique, y compris les données de configuration actuelles.
- [set-type-default-version](#) pour spécifier la version d'une extension qui est la version par défaut.
- [deactivate-type](#) pour désactiver un module ou un type de ressource tiers public précédemment activé dans votre compte.

Activation d'une extension publique tierce dans votre compte

La rubrique suivante vous explique comment activer une extension publique tierce dans votre compte, ce qui la rend utilisable dans le compte et la région où elle a été activée.

 Note

Avant de poursuivre, assurez-vous d'avoir créé l'extension [Rôle IAM](#) que vous utiliserez.

Rubriques

- [Activation d'une extension publique \(console\)](#)
- [Activer une extension publique \(AWS CLI\)](#)

Activation d'une extension publique (console)

Suivez les étapes de cette section pour utiliser la console afin de :

- Activation d'une extension publique tierce
- Spécifiez des données de configuration d'extension supplémentaires pour votre compte

Pour activer une extension publique à utiliser dans votre compte

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Dans le volet de navigation, sous Registre, sélectionnez Extensions publiques.
4. Utilisez le filtre pour choisir le type d'extension, puis sélectionnez Tiers. (Les extensions publiées par AWS sont activées par défaut.)
5. Sélectionnez l'extension, puis choisissez Activer.

Si plusieurs versions d'une extension sont disponibles, vous pouvez utiliser le menu Version pour choisir la version de l'extension que vous souhaitez activer. La version par défaut est la plus actuelle.

6. Pour le nom de l'extension, vous pouvez soit conserver l'option Utiliser la valeur par défaut, soit choisir Remplacer la valeur par défaut, puis saisir le type d'extension que vous souhaitez utiliser avec cette extension. L'alias doit respecter le format recommandé pour le type d'extension. Pour de plus amples informations, veuillez consulter [Utilisez des alias pour faire référence aux extensions](#).

7. Si l'extension que vous activez est un Hook ou un type de ressource, pour l'ARN du rôle d'exécution, spécifiez le rôle IAM CloudFormation à assumer lors de l'appel de l'extension. Pour de plus amples informations, veuillez consulter [Configurer un rôle d'exécution avec des autorisations IAM et une politique de confiance pour l'accès public aux extensions](#).
8. Pour la configuration de la journalisation, spécifiez les informations de configuration de la journalisation pour une extension, si vous le souhaitez. Par exemple :

```
{  
  "logRoleArn": "arn:aws:iam::account:role/rolename",  
  "logGroupName": "log-group-name"  
}
```

Les informations de configuration de la journalisation ne sont pas requises, mais recommandées à des fins de débogage. Pour utiliser la configuration de la journalisation avec des hooks, ajoutez la même politique d'approbation que le rôle d'exécution spécifié, afin que le rôle du journal puisse écrire des journaux dans votre groupe de journaux.

logRoleArn et logGroupName les noms de clés sont sensibles à la casse.

9. Pour la gestion des versions, Mises à jour automatiques, choisissez le mode de réception des mises à jour.
 - On (Activé) : met automatiquement à jour vers la dernière version mineure. Les versions majeures sont mises à jour manuellement.
 - On (Désactivé) : ne met jamais automatiquement à jour vers la dernière version. Toutes les versions sont mises à jour manuellement.

Pour de plus amples informations, veuillez consulter [Utiliser automatiquement les nouvelles versions des extensions](#).

Si l'extension nécessite une configuration supplémentaire, vous avez la possibilité de spécifier les données de configuration maintenant ou après l'activation de l'extension.

Important

Si l'extension que vous activez est un Hook, cette étape est obligatoire. Vous devez spécifier `ENABLED` pour la propriété `HookInvocationStatus`. Cette opération active les propriétés du hook définies dans la section `properties` de schéma du hook. Pour plus d'informations,

consultez la [référence de syntaxe du schéma de configuration Hook](#) dans le guide de l'utilisateur de Hooks CloudFormation .

Pour spécifier les données de configuration

1. Pour Configuration, choisissez Configurer maintenant, puis sélectionnez Activer l'extension.

CloudFormation affiche la page Configurer l'extension. Pour afficher le schéma de configuration actuel de l'extension, assurez-vous d'activer View configuration schema (Afficher le schéma de configuration).
2. Dans la zone de texte Fichier JSON de configuration, saisissez une chaîne JSON qui représente les données de configuration que vous souhaitez spécifier pour cette extension. Le JSON que vous spécifiez doit être validé par rapport au schéma de configuration de l'extension.
3. Sélectionnez Configurer l'extension.

Si vous préférez configurer l'extension après l'activation, vous pouvez ignorer cette étape et fournir les données de configuration ultérieurement.

1. Pour Configuration, choisissez Configurer ultérieurement, puis sélectionnez Activer l'extension.
2. Une fois l'extension activée, vous pouvez la configurer en accédant à la page des extensions activées et en fournissant les données de configuration.

Activer une extension publique (AWS CLI)

Suivez les étapes décrites dans cette section pour utiliser le AWS CLI pour :

- Activation d'une extension publique tierce
- Spécifiez des données de configuration d'extension supplémentaires pour votre compte

Activation des Hooks publics

En activant les Hooks dans votre compte, vous autorisez un Hook à utiliser les autorisations définies par votre Compte AWS. CloudFormation supprime les autorisations non requises avant de les transmettre au Hook. CloudFormation recommande aux clients ou aux utilisateurs de Hook de consulter les autorisations Hook et de connaître les autorisations auxquelles les Hooks sont autorisés avant d'activer Hooks dans votre compte.

Pour activer un Hook public à utiliser dans votre compte (AWS CLI)

1. Obtenez l'ARN pour votre Hook et enregistrez-le. Vous pouvez obtenir l'ARN d'un Hook en utilisant le AWS Management Console ou AWS CLI. Pour de plus amples informations, veuillez consulter [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Utilisez la commande [set-type-configuration](#) pour spécifier les données de configuration. Le JSON que vous transmettez à `--configuration` doit être validé par rapport au schéma de configuration du Hook. Pour activer le Hook pour toutes les opérations de stack, vous devez définir la propriété `HookInvocationStatus` sur `ENABLED` dans la section `HookConfiguration`.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Pour plus d'informations sur les options de configuration `HookConfiguration`, consultez la [référence de syntaxe du schéma de configuration Hook](#) dans le Guide de l'utilisateur de Hooks CloudFormation .

Activer les modules publics et les types de ressources

Pour activer une extension publique à utiliser dans votre compte (AWS CLI)

- Utilisez la commande [activate-type](#) pour activer l'extension et spécifier s'il faut la mettre à jour chaque fois qu'une nouvelle version mineure de l'extension est publiée.

L'exemple ci-dessous spécifie l'Amazon Resource Name (ARN) public d'une extension publique à activer pour ce compte. En outre, il précise que l'extension est CloudFormation mise à jour chaque fois qu'une nouvelle version mineure est publiée.

```
aws cloudformation activate-type \  
  --public-type-arn public_extension_ARN \  
  --execution-role-arn arn:aws:iam::123456789012:role/my-execution-role \  
  --auto-update true --region us-west-2
```


Cette commande renvoie un ARN de l'extension activée.

```
{
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example"
}
```

Mettre à jour la version d'une extension publique (AWS CLI)

Utilisez [activate-type](#) pour réactiver l'extension.

Utilisez cette option `--version-bump` pour spécifier si vous souhaitez mettre à jour l'extension vers la dernière version MAJOR ou la version MINOR la plus récente.

```
aws cloudformation activate-type --type RESOURCE \
--type-name Example::Test::1234567890abcdef0 \
--type-name-alias Example::Test::Alias \
--version-bump MAJOR --region us-west-2
```

Mettre à jour une extension tierce publique dans votre compte

Après avoir activé une extension publique tierce, vous pouvez mettre à jour la plupart des détails de l'extension depuis votre compte.

Pour mettre à jour une extension publique dans votre compte (console)

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Dans le volet de navigation, sous Registre, sélectionnez Extensions activées.
4. Recherchez l'extension que vous souhaitez mettre à jour et sélectionnez-la. Pour de plus amples informations, veuillez consulter [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).
5. Dans le menu Actions, sélectionnez Modifier, puis l'option d'édition appropriée :
 - Pour mettre à jour le schéma de configuration, consultez [Modifier les données de configuration des extensions de votre compte](#).

- Pour activer ou désactiver les mises à jour automatiques :
 1. Choisissez Modifier les mises à jour automatiques.
 2. Sélectionnez Activé ou Désactivé, puis cliquez sur Enregistrer. Pour de plus amples informations, veuillez consulter [Utiliser automatiquement les nouvelles versions des extensions](#).
- Pour mettre à jour le rôle d'exécution :
 1. Sélectionnez Modifier le rôle d'exécution.
 2. Spécifiez l'ARN du rôle IAM que vous CloudFormation souhaitez utiliser lors de l'appel de cette extension, puis choisissez Enregistrer. Pour de plus amples informations, veuillez consulter [Configurer un rôle d'exécution avec des autorisations IAM et une politique de confiance pour l'accès public aux extensions](#).
- Pour mettre à jour la configuration de la journalisation :
 1. Sélectionnez Modifier la configuration de la journalisation.
 2. Modifiez la configuration de journalisation JSON, puis sélectionnez Enregistrer.

Désactivation des extensions publiques tierces dans votre compte

Lorsque vous n'avez plus besoin d'une extension publique tierce activée, utilisez les procédures suivantes pour la désactiver dans votre compte.

Rubriques

- [Désactiver une extension publique dans votre compte \(console\)](#)
- [Désactiver une extension publique dans votre compte \(AWS CLI\)](#)
- [Désactiver un Hook dans votre compte \(AWS CLI\)](#)

Désactiver une extension publique dans votre compte (console)

Pour désactiver une extension publique dans votre compte

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Dans le volet de navigation, sous Registre, sélectionnez Extensions activées.

4. Trouvez l'extension que vous souhaitez désactiver et sélectionnez-la. Pour de plus amples informations, veuillez consulter [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).
5. Dans le menu Actions, sélectionnez Désactiver.
6. Sélectionnez Désactiver.

Désactiver une extension publique dans votre compte (AWS CLI)

Utilisez la commande [deactivate-type](#) suivante.

```
aws cloudformation deactivate-type --type MODULE \  
  --type-name Example::Test::Type::MODULE \  
  --region us-west-2
```

Désactiver un Hook dans votre compte (AWS CLI)

La désactivation d'un Hook empêche le Hook de s'exécuter dans le vôtre Compte AWS sans le supprimer.

Utilisez la commande [set-type-configuration](#) et spécifiez HookInvocationStatus en tant que DISABLED pour désactiver un Hook.

L'exemple suivant indique le nom Région AWS et l'Amazon Resource Name (ARN) du Hook qui est désactivé.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "DISABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn "arn:aws:cloudformation:us-west-2:123456789012:type/hook/MyTestHook" --  
  region us-west-2
```

Pour plus d'informations, voir [Désactiver et activer les CloudFormation Hooks](#) dans le Guide de l'utilisateur CloudFormation des Hooks.

Utilisez des extensions privées tierces qui ont été partagées avec vous

Pour utiliser des extensions privées tierces qui ont été partagées avec vous, vous devez d'abord les enregistrer auprès des comptes et des régions dans lesquels vous souhaitez les utiliser. CloudFormation L'enregistrement de l'extension en télécharge une copie dans le CloudFormation registre de votre compte et l'active. Une fois que vous avez enregistré une extension privée, elle apparaît dans le CloudFormation registre correspondant à celle-ci Compte AWS et à la région, et vous pouvez l'utiliser dans vos modèles de stack.

Rubriques

- [Autorisations IAM pour l'enregistrement d'une extension privée tierce](#)
- [AWS CLI Commandes couramment utilisées pour travailler avec des extensions privées](#)
- [Enregistrez une extension privée tierce dans votre compte](#)
- [Supprimer les extensions privées tierces de votre compte](#)

Autorisations IAM pour l'enregistrement d'une extension privée tierce

Dans le cadre de l'enregistrement d'une extension privée, vous pouvez spécifier un compartiment Amazon S3 contenant le package du projet d'extension. Ce package contient tous les fichiers sources nécessaires à l'extension que vous souhaitez enregistrer. L'utilisateur qui enregistre l'extension doit pouvoir accéder au package du projet dans ce compartiment Amazon S3. Pour ce faire, l'utilisateur doit disposer d'autorisations [GetObject](#) pour le package d'extension.

Cela est vrai que vous utilisiez la [register-type](#) commande ou AWS CLI la [submit](#) commande de la CloudFormation CLI.

Pour plus d'informations, consultez [Actions, ressources et clés de condition pour Amazon S3](#) dans la Référence de l'autorisation de service.

AWS CLI Commandes couramment utilisées pour travailler avec des extensions privées

Les commandes couramment utilisées pour travailler avec les extensions privées comprennent :

- [register-type](#) pour enregistrer une extension privée dans votre compte.
- [describe-type-registration](#) pour renvoyer l'état actuel d'une demande d'enregistrement.

- [list-types](#) pour répertorier les extensions de votre compte.
- [describe-type](#) pour renvoyer des informations détaillées sur une extension ou une version d'extension spécifique, y compris les données de configuration actuelles.
- [set-type-configuration](#) pour spécifier les données de configuration d'une extension dans votre compte et pour désactiver et activer les Hooks.
- [set-type-default-version](#) pour spécifier la version d'une extension qui est la version par défaut.
- [deregister-type](#) pour supprimer une extension privée ou une version d'extension de votre compte.

Enregistrez une extension privée tierce dans votre compte

Cette rubrique décrit les étapes à suivre pour enregistrer une extension privée tierce qui vous a été partagée afin qu'elle soit disponible dans votre compte.

Note

Avant de continuer, vérifiez que vous disposez des [Autorisations IAM](#) nécessaires pour enregistrer une extension privée.

Pour enregistrer une extension privée partagée avec vous (AWS CLI)

1. Recherchez le compartiment Amazon S3 qui contient le package de projet pour l'extension privée que vous souhaitez enregistrer dans votre compte.
2. Utilisez la commande [register-type](#) pour enregistrer l'extension privée dans votre compte.

Par exemple, la commande suivante enregistre le type de ressource `My::Resource::Example` dans le fichier spécifié Compte AWS.

```
aws cloudformation register-type --type RESOURCE \  
  --type-name My::Resource::Example \  
  --schema-handler-package [s3 object path] --region us-west-2
```

RegisterType est une opération asynchrone. Elle renvoie un jeton d'enregistrement que vous pouvez utiliser pour suivre la progression de votre demande d'enregistrement.

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"
```

```
}
```

Si votre extension appelle dans AWS APIs le cadre de ses fonctionnalités, vous devez créer un rôle d'exécution IAM incluant les autorisations nécessaires pour les appeler AWS APIs, et fournir ce rôle d'exécution dans votre compte. Vous pouvez ensuite spécifier ce rôle d'exécution à l'aide de l'`--execution-role-arn` option. CloudFormation assume ensuite ce rôle d'exécution pour fournir à votre type de ressource les informations d'identification appropriées.

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (Facultatif) Utilisez le jeton d'enregistrement avec la commande [describe-type-registration](#) pour suivre la progression de votre demande d'enregistrement.

Lorsque la demande d'enregistrement est CloudFormation terminée, l'état d'avancement de la demande est défini sur `COMPLETE`.

L'exemple suivant utilise le jeton d'enregistrement renvoyé par la commande `describe-type-registration` ci-dessus pour renvoyer les informations relatives à l'état de l'enregistrement.

```
aws cloudformation describe-type-registration \
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \
  --region us-west-2
```

La commande renvoie le résultat suivant.

```
{
  "ProgressStatus": "COMPLETE",
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example",
  "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example/00000001"
}
```

Important

Si l'extension que vous enregistrez est un Hook, l'étape suivante est obligatoire. Vous devez spécifier `ENABLED` pour la propriété `HookInvocationStatus`. Cette opération active les propriétés du hook définies dans la section `properties` de schéma du hook. Pour plus

d'informations, consultez la [référence de syntaxe du schéma de configuration Hook](#) dans le guide de l'utilisateur de Hooks CloudFormation .

Pour spécifier les données de configuration d'un Hook (AWS CLI)

1. Obtenez l'ARN pour votre Hook et enregistrez-le. Vous pouvez obtenir l'ARN d'un Hook en utilisant le AWS Management Console ou AWS CLI. Pour de plus amples informations, veuillez consulter [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Utilisez la commande [set-type-configuration](#) pour spécifier les données de configuration. Le JSON que vous transmettez à `--configuration` doit être validé par rapport au schéma de configuration du Hook. Pour activer le Hook, vous devez définir la propriété `HookInvocationStatus` sur `ENABLED` dans la section `HookConfiguration`.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Pour plus d'informations, consultez la [référence de syntaxe du schéma de configuration Hook](#) dans le guide de l'utilisateur de Hooks CloudFormation .

Supprimer les extensions privées tierces de votre compte

Pour supprimer une extension privée tierce ou une version d'extension, utilisez la commande [deregister-type](#).

Vous pouvez annuler l'enregistrement d'une version d'extension spécifique ou de l'extension dans son ensemble. Pour désenregistrer une extension, vous devez désenregistrer individuellement toutes les versions enregistrées de cette extension. Si une extension ne dispose que d'une seule version enregistrée, la désinscription de cette version entraîne la désinscription de l'extension elle-même. Vous ne pouvez pas désenregistrer la version par défaut d'une extension, sauf s'il s'agit de la seule version enregistrée de cette extension, auquel cas l'extension elle-même est également désenregistrée.

 Warning

Annuler l'enregistrement d'une extension privée ne peut pas être annulé. Cette action permettra de :

- Rendez l'extension inutilisable dans toutes les CloudFormation opérations.
- Provoquer des échecs dans les futures mises à jour de la pile qui utilisent cette extension (pour les modules et les types de ressources). Bien que vous puissiez réenregistrer l'extension en privé ultérieurement, cela peut entraîner des échecs si vous CloudFormation utilisez une version antérieure.

Avant de continuer, utilisez les commandes [list-stacks](#) et [get-template](#) pour vérifier qu'aucune pile active n'utilise cette extension.

Exemples de commandes de désenregistrement d'extension

Cette section fournit des exemples illustrant les différentes façons de désenregistrer des extensions privées.

Désenregistrer par nom de type

Utilisez la commande [deregister-type](#) avec les options `--type` et `--type-name` pour désenregistrer votre extension.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE
```

Désenregistrer par type, nom et version

Pour désenregistrer une version spécifique de votre extension, spécifiez l'option `--version-id` dans la commande.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --version-id 00000001
```


 Tip

Pour définir d'abord une autre version de l'extension par défaut, utilisez la [set-type-default-version](#) commande.

Annuler l'enregistrement par ARN

Utilisez l'option `--arn` et spécifiez l'ARN de votre extension pour la désenregistrer.

```
aws cloudformation deregister-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource
```

Modifier les données de configuration des extensions de votre compte

Cette rubrique fournit des conseils sur la modification des données de configuration des extensions de votre compte dans une région spécifique. Les extensions peuvent inclure des propriétés de configuration qui s'appliquent à toutes les instances de l'extension pour un compte et une région donnés. Ils sont définis par l'auteur de l'extension dans la définition de configuration de l'extension. Si certaines propriétés sont requises dans la définition de configuration de l'extension, vous devez les spécifier avant de pouvoir utiliser l'extension dans votre compte et votre région.

Pour plus d'informations sur la manière dont les définitions de configuration sont définies lors du développement d'une extension, consultez la documentation suivante.

- [Référence syntaxique du schéma de configuration Hook](#)
- [Définition de la configuration d'une extension au niveau du compte](#)

Rubriques

- [Autorisations requises pour utiliser des références dynamiques](#)
- [Modifier les données de configuration d'une extension \(console\)](#)
- [Modifier les données de configuration d'une extension \(AWS CLI\)](#)

Autorisations requises pour utiliser des références dynamiques

Si vos données de configuration incluent des références dynamiques à des valeurs stockées dans AWS Systems Manager ou AWS Secrets Manager, tout rôle utilisé pour provisionner le type (par exemple, lors de la création ou de la mise à jour d'une pile) doit disposer des autorisations appropriées pour récupérer cette valeur. En particulier :

- Si les données de configuration contiennent un paramètre stocké dans le Parameter Store AWS Systems Manager, l'utilisateur ou le rôle utilisé pour provisionner le type doit disposer des autorisations nécessaires pour appeler [GetParameter](#).
- Si les données de configuration contiennent un secret stocké dans AWS Secrets Manager, l'utilisateur ou le rôle utilisé pour configurer le type doit être autorisé à appeler [GetSecretValue](#).

Pour de plus amples informations, veuillez consulter [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

Modifier les données de configuration d'une extension (console)

Suivez les étapes de cette section pour utiliser la console afin de :

- Afficher les données de configuration actuelles d'une extension
- Mettre à jour les données de configuration de l'extension pour votre compte

Pour afficher les données de configuration actuelles d'une extension

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Dans le volet de navigation, sous Registre, sélectionnez Extensions activées.
4. Recherchez l'extension que vous souhaitez afficher. Pour de plus amples informations, veuillez consulter [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).
5. Choisissez l'extension pour afficher ses détails.
6. Sur la page des détails de l'extension, choisissez l'onglet Configuration.
7. Développez l'onglet Configuration schema (Schéma de configuration) pour voir le schéma de configuration défini pour l'extension.

8. Développez l'onglet Configuration pour voir la configuration actuelle que vous avez définie pour cette extension.

Pour mettre à jour les données de configuration d'une extension

1. Sur la page des détails de l'extension, dans l'onglet Configuration, sélectionnez Modifier la configuration.

Vous pouvez également, dans Actions, sélectionner Modifier, puis Modifier la configuration.

CloudFormation affiche la page Configurer l'extension. Vérifiez que View configuration schema (Afficher un schéma de configuration) est activé pour voir le schéma de définition de la configuration actuelle de l'extension.

2. Dans la zone de texte Configuration JSON (Fichier JSON de configuration), saisissez une chaîne JSON qui représente le schéma de configuration que vous souhaitez définir pour cette extension. Il doit être conforme au schéma défini dans Configuration schema (Schéma de configuration).
3. Sélectionnez Configurer l'extension.

Modifier les données de configuration d'une extension (AWS CLI)

Suivez les étapes décrites dans cette section pour utiliser le AWS CLI pour :

- Afficher les données de configuration actuelles d'une extension
- Mettre à jour les données de configuration de l'extension pour votre compte

Pour afficher les données de configuration actuelles d'une extension

- Utilisez la commande [describe-type](#) pour obtenir des informations détaillées sur l'extension. L'élément ConfigurationSchema de la sortie contient la définition de la configuration actuelle de l'extension dans une région donnée.

Vous pouvez également utiliser la commande [batch-describe-type-configurations](#) pour renvoyer les données de configuration relatives à plusieurs extensions.

Pour mettre à jour les données de configuration d'une extension

- Utilisez la commande [set-type-configuration](#) pour spécifier les données de configuration. Le fichier JSON que vous transmettez pour `--configuration` doit être conforme au schéma de configuration de l'extension.

Dans l'exemple suivant, la commande `set-type-configuration` spécifie les données de configuration `{"CredentialKey": "testUserCredential"}` pour l'option `--configuration`.

```
aws cloudformation set-type-configuration --type RESOURCE \  
  --type-name My::Resource::Example \  
  --configuration-alias default \  
  --configuration '{"CredentialKey": "testUserCredential"}' \  
  --region us-west-2
```

Enregistrez les types de ressources dans AWS Config

Vous pouvez spécifier un suivi AWS Config automatique de vos types de ressources privées et enregistrer les modifications apportées à ces ressources en tant qu'éléments de configuration. Cela vous permet de consulter l'historique de configuration de ces types de ressources privées, ainsi que d'écrire des AWS Config Rules règles pour vérifier les meilleures pratiques de configuration. AWS Config est requis pour l'extension Hook.

Pour suivre AWS Config automatiquement vos types de ressources privées :

- Gérez les ressources via CloudFormation. Cela inclut l'exécution de toutes les opérations de création, de mise à jour et de suppression de ressources via CloudFormation.

Note

Si vous utilisez un rôle IAM pour effectuer vos opérations de stack, ce rôle IAM doit être autorisé à effectuer les actions suivantes : AWS Config

- [PutResourceConfig](#)
- [DeleteResourceConfig](#)

- Configurez AWS Config pour enregistrer tous les types de ressources. Pour plus d'informations, consultez la section [Enregistrement des configurations pour les ressources tierces à l'aide du AWS CLI](#) dans le Guide du développeur AWS Config .

Note

AWS Config ne prend pas en charge l'enregistrement de ressources privées contenant des propriétés définies à la fois comme obligatoires et en écriture seule.

De par leur conception, les propriétés des ressources définies en écriture seule ne sont pas renvoyées dans le schéma utilisé pour créer l'élément de AWS Config configuration. Pour cette raison, l'inclusion d'une propriété définie comme étant à la fois en écriture seule et obligatoire entraînera l'échec de la création de l'élément de configuration, car une propriété requise ne sera pas présente. Pour afficher le schéma qui sera utilisé pour créer l'élément de configuration, vous pouvez consulter la propriété `schema` de l'action [DescribeType](#).

Pour plus d'informations sur les éléments de configuration, veuillez consulter la rubrique [Éléments de configuration](#) dans le Guide du développeur AWS Config .

Prévention de l'enregistrement de propriétés sensibles dans un élément de configuration

Votre type de ressource peut contenir des propriétés que vous considérez comme étant des informations sensibles (telles que des mots de passe, des secrets ou d'autres données sensibles) que vous ne souhaitez pas enregistrer dans l'élément de configuration. Pour prévenir l'enregistrement d'une propriété dans l'élément de configuration, vous pouvez inclure cette propriété dans la liste `writeOnlyProperties` de votre schéma de type de ressource. Les propriétés de ressources répertoriées en tant que `writeOnlyProperties` peuvent être spécifiées par l'utilisateur, mais ne seront pas renvoyées par une demande `read` ou `list`.

Pour plus d'informations, consultez le guide [writeOnlyProperties](#) de l'utilisateur de la CloudFormation CLI.

Livraison continue avec CodePipeline

La livraison continue est une pratique de publication dans laquelle les modifications du code sont automatiquement générées, testées et préparées en vue d'une publication en production. Avec CloudFormation et CodePipeline, vous pouvez utiliser la livraison continue pour créer et tester automatiquement les modifications apportées à vos CloudFormation modèles avant de les promouvoir en production. Ce processus de publication vous permet d'apporter des modifications rapides et fiables à votre AWS infrastructure.

Par exemple, vous pouvez créer un flux de travail qui crée automatiquement une pile de test lorsque vous soumettez un modèle mis à jour à un référentiel de code. Après CloudFormation avoir créé la pile de test, vous pouvez la tester, puis décider d'appliquer les modifications à une pile de production. Pour plus d'informations sur les avantages de la livraison continue, consultez [Qu'est-ce que la livraison continue ?](#).

CodePipeline À utiliser pour créer un flux de livraison continue en créant un pipeline pour les CloudFormation piles. CodePipeline est intégré à CloudFormation, ce qui vous permet de spécifier CloudFormation des actions spécifiques, telles que la création, la mise à jour ou la suppression d'une pile, dans un pipeline. Pour plus d'informations CodePipeline, consultez le [guide de AWS CodePipeline l'utilisateur](#).

Rubriques

- [Procédure : Création d'un pipeline pour les piles de test et de production](#)
- [CloudFormation référence aux propriétés de configuration](#)
- [CloudFormation artefacts](#)
- [Utilisation de fonctions de remplacement de paramètres avec des pipelines CodePipeline](#)

Procédure : Création d'un pipeline pour les piles de test et de production

Imaginez un processus de publication dans lequel vous soumettez un CloudFormation modèle, qui CloudFormation est ensuite utilisé pour créer automatiquement une pile de tests. Après avoir passé en revue la pile de test, vous pouvez obtenir un aperçu de la façon dont vos modifications vont changer votre pile de production. Vous choisissez alors de procéder ou non à leur implémentation. Pour réaliser ce flux de travail, vous pouvez CloudFormation créer votre pile de tests, supprimer

la pile de tests, créer un ensemble de modifications, puis exécuter l'ensemble de modifications. Cependant, à chaque action, vous devez interagir manuellement avec CloudFormation. Dans cette présentation détaillée, nous allons créer un CodePipeline pipeline qui automatise bon nombre de ces actions, afin de vous aider à mettre en place un flux de travail de livraison continu avec vos CloudFormation stacks.

Conditions préalables

Cette procédure pas à pas suppose que vous avez utilisé CodePipeline et que vous savez comment fonctionnent les pipelines CloudFormation, les AWS CloudFormation modèles et les piles. Pour plus d'informations CodePipeline, consultez le [guide de AWS CodePipeline l'utilisateur](#). Vous devez également avoir un compartiment Amazon S3 dans le compartiment dans Région AWS lequel vous créez votre pipeline.

Important

L'exemple de WordPress modèle crée une instance EC2 qui nécessite une connexion à Internet. Vérifiez que votre VPC et votre sous-réseau par défaut autorisent le trafic vers Internet.

Présentation de la procédure

Cette procédure pas à pas crée un pipeline pour un exemple de WordPress site dans une pile. Le pipeline est divisé en trois étapes. Chaque étape doit contenir au moins une action, qui constitue une tâche que le pipeline effectue sur vos artefacts (vos entrées). Une étape organise des actions dans un pipeline. CodePipeline doit terminer toutes les actions d'une étape avant que l'étape ne traite de nouveaux artefacts ; par exemple, si vous avez envoyé une nouvelle entrée pour exécuter de nouveau le pipeline.

A la fin de cette procédure, vous disposerez d'un pipeline qui réalise le flux de travail suivant :

1. La première étape du pipeline récupère un artefact source (un CloudFormation modèle et ses fichiers de configuration) dans un référentiel.

Vous allez préparer un artefact qui inclut un exemple de WordPress modèle et le télécharger dans un compartiment S3.

2. Dans la deuxième étape, le pipeline crée une pile de test, puis attend votre approbation.

Après avoir passé en revue la pile de test, vous pouvez choisir de continuer avec le pipeline d'origine ou créer et soumettre un autre artefact pour apporter des modifications. Si vous l'approuvez, cette étape supprime la pile de test et le pipeline passe à l'étape suivante.

3. Dans la troisième étape, le pipeline crée un jeu de modifications par rapport à la pile de production, puis attend votre approbation.

Dans votre exécution initiale, vous ne disposerez pas encore de pile de production. L'ensemble de modifications vous indique toutes les ressources qui CloudFormation seront créées. Si vous l'approuvez, cette étape exécute le jeu de modifications et crée votre pile de production.

Note

CloudFormation est un service gratuit. Toutefois, les AWS ressources, telles que l'instance EC2, que vous incluez dans votre pile vous sont facturées au tarif en vigueur pour chacune d'entre elles. Pour plus d'informations sur AWS les prix, consultez la page détaillée de chaque produit à l'[adresse http://aws.amazon.com](http://aws.amazon.com).

Étape 1 : Modifier l'artefact et le charger dans un compartiment S3

Avant de créer votre pipeline, vous devez configurer votre référentiel source et vos fichiers. CodePipeline copie ces fichiers source dans le [magasin d'artefacts](#) de votre pipeline, puis les utilise pour effectuer des actions dans votre pipeline, telles que la création d'une CloudFormation pile.

Lorsque vous utilisez Amazon Simple Storage Service (Amazon S3) comme référentiel source CodePipeline, vous devez compresser vos fichiers sources avant de les télécharger dans un compartiment S3. Le fichier compressé est un CodePipeline artefact qui peut contenir un CloudFormation modèle, un fichier de configuration de modèle ou les deux. Nous fournissons un artefact qui contient un exemple de WordPress modèle et deux fichiers de configuration de modèle. Les deux fichiers de configuration spécifient les valeurs des paramètres du WordPress modèle. CodePipeline les utilise lorsqu'il crée les WordPress piles. L'un des fichiers contient les valeurs des paramètres pour une pile de test et l'autre fichier pour une pile de production. Vous aurez besoin de modifier les fichiers de configuration, par exemple, pour spécifier le nom d'une paire de clés EC2 que vous possédez déjà. Pour plus d'informations sur les artefacts, consultez [CloudFormation artefacts](#).

Après avoir créé votre artefact, vous allez le charger dans un compartiment S3.

Pour modifier et charger l'artefact

1. Téléchargez et ouvrez l'exemple d'artefact : <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/wordpress-single-instance.zip>.

L'artefact contient trois fichiers :

- Exemple de WordPress modèle : `wordpress-single-instance.yaml`
 - Le fichier de configuration du modèle pour la pile de test : `test-stack-configuration.json`
 - Le fichier de configuration du modèle pour la pile de production : `prod-stack-configuration.json`
2. Extrayez tous les fichiers et utilisez n'importe quel éditeur de texte pour modifier les fichiers de configuration du modèle.

Ouvrez les fichiers de configuration pour vérifier qu'ils contiennent des paires clé-valeur correspondant aux paramètres du WordPress modèle. Les fichiers de configuration spécifient les valeurs des paramètres que votre pipeline utilise lorsqu'il crée les piles de test et de production.

Modifiez le fichier `test-stack-configuration.json` pour spécifier les valeurs des paramètres pour la pile de test et le fichier `prod-stack-configuration.json` pour la pile de production.

- Remplacez les valeurs des `DBRootPassword` touches `DBPassword` et par des mots de passe que vous pouvez utiliser pour vous connecter à votre WordPress base de données. Comme défini dans le WordPress modèle, les valeurs des paramètres ne doivent contenir que des caractères alphanumériques.
 - Remplacez la valeur de la clé `KeyName` par un nom de paire de clés EC2 existant dans la région dans laquelle vous allez créer votre pipeline.
3. Ajoutez les fichiers de configuration modifiés au fichier de l'artefact d'origine (`.zip`) en remplaçant les fichiers en double.

Vous disposez maintenant d'un artefact personnalisé que vous pouvez charger dans un compartiment S3.

4. [Chargez l'artefact dans un compartiment S3 que vous possédez.](#)

Notez l'emplacement du fichier. Vous devrez spécifier l'emplacement de ce fichier lors de la création de votre pipeline.

Remarques sur l'artefact et le compartiment S3 :

- Utilisez un bucket situé dans la même AWS région que celle dans laquelle vous allez créer votre pipeline.
- CodePipeline nécessite que la gestion des [versions du bucket soit activée](#).
- Vous pouvez également utiliser des services qui ne vous obligent pas à compresser vos fichiers avant de les télécharger, GitHub par exemple pour votre dépôt source. CodeCommit
- Les artefacts peuvent contenir des informations sensibles telles que des mots de passe. Limitez l'accès de façon à ce que seuls les utilisateurs autorisés puissent consulter le fichier. Lorsque vous le faites, assurez-vous qu'il CodePipeline peut toujours accéder au fichier. Par exemple, si vous chargez votre artefact dans un compartiment S3, utilisez [des politiques de compartiment S3 ou des politiques utilisateur](#) pour en limiter l'accès.

Vous disposez désormais d'un artefact que CodePipeline peut être intégré à votre pipeline. À l'étape suivante, vous allez spécifier l'emplacement de l'artefact et créer le WordPress pipeline.

Étape 2 : Créer la pile du pipeline

Pour créer le WordPress pipeline, vous allez utiliser un exemple de CloudFormation modèle. Outre la création du pipeline, le modèle définit des rôles de service Gestion des identités et des accès AWS (IAM) pour CodePipeline et CloudFormation, un compartiment S3 pour le magasin d' CodePipeline artefacts et une rubrique Amazon Simple Notification Service (Amazon SNS) à laquelle le pipeline envoie des notifications, telles que des notifications concernant les avis. L'exemple de modèle facilite le provisionnement et la configuration de ces ressources dans une seule CloudFormation pile.

Pour plus d'informations sur la configuration du pipeline, consultez [Fonctions du pipeline](#).

Important

L'exemple de WordPress modèle crée une instance EC2 qui nécessite une connexion à Internet. Vérifiez que votre VPC et votre sous-réseau par défaut autorisent le trafic vers Internet.

Pour créer la pile du pipeline

1. Téléchargez le modèle d'exemple sur <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic-pipeline.yml>. Enregistrez-le sur votre ordinateur.
2. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation/>.
3. Choisissez une AWS région qui prend en charge CodePipeline et CloudFormation.

Pour plus d'informations, consultez les sections [CodePipeline Points de terminaison et quotas](#) et [CloudFormation Points de terminaison et quotas](#) dans le. Références générales AWS

4. Sélectionnez Créer la pile.
5. Sous Specify template (Spécifier un modèle), choisissez Upload a template file (Charger un fichier de modèle), puis choisissez le modèle que vous venez de télécharger, basic-pipeline.yml.
6. Choisissez Suivant.
7. Pour le Nom de la pile, tapez sample-WordPress-pipeline.
8. Dans la section Paramètres, spécifiez les valeurs suivantes pour les paramètres, puis choisissez Suivant. Lorsque vous définissez les paramètres de pile, si vous avez conservé les mêmes noms pour le WordPress modèle et ses fichiers de configuration, vous pouvez utiliser les valeurs par défaut. Dans le cas contraire, spécifiez les noms de fichiers utilisés.

PipelineName

Nom de votre pipeline, comme WordPress-test-pipeline.

S3Bucket

Nom du compartiment S3 dans lequel vous avez enregistré votre artefact (fichier .zip).

SourceS3Key

Nom de fichier de votre artefact. Si vous avez enregistré l'artefact dans un dossier, incluez-le dans le nom de fichier, comme folder/subfolder/wordpress-single-instance.zip.

E-mail

Adresse e-mail à laquelle CodePipeline envoie la notification du pipeline, telle que myemail@example.com.

9. Dans cette procédure détaillée, vous n'avez pas besoin d'ajouter des balises ni de spécifier des paramètres avancés. Dès lors, cliquez sur Suivant.

10. Assurez-vous que le nom de la pile et l'URL du modèle sont corrects, puis choisissez Créer une pile.
11. Pour confirmer que vous savez que des ressources IAM CloudFormation peuvent être créées, cochez la case.

La création de votre pile peut prendre plusieurs minutes. AWS CloudFormation Pour surveiller la progression, affichez les événements de la pile. Pour de plus amples informations, veuillez consulter [Surveiller la progression de la pile](#).

Une fois votre stack créé, CodePipeline démarre votre nouveau pipeline. Pour afficher son état, consultez la [console CodePipeline](#). Dans la liste des pipelines, choisissez WordPress-test-pipeline.

Fonctions du pipeline

Cette section explique les trois étapes du pipeline, à l'aide d'extraits de l'exemple de modèle de WordPress pipeline.

Étape 1 : Source

La première étape du pipeline concerne la source : vous spécifiez l'emplacement de votre code source. Chaque fois que vous envoyez une révision à cet emplacement, votre pipeline est CodePipeline réexécuté.

Le code source, identifié par son nom de fichier, se trouve dans un compartiment S3. Vous avez spécifié ces valeurs en tant que valeurs du paramètre d'entrée lorsque vous avez créé la pile du pipeline. Pour permettre l'utilisation de l'artefact source dans les étapes suivantes, l'extrait spécifie la propriété `OutputArtifacts` avec le nom `TemplateSource`. Pour utiliser cet artefact dans les étapes suivantes, spécifiez `TemplateSource` comme artefact d'entrée.

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
```

```
S3ObjectKey: !Ref 'SourceS3Key'
OutputArtifacts:
  - Name: TemplateSource
```

Étape 2 : TestStage

A l'étape TestStage, le pipeline crée la pile de test, attend l'approbation, puis supprime la pile de test.

Pour l'CreateStackaction, le pipeline utilise le fichier de configuration de test et le WordPress modèle pour créer la pile de tests. Les deux fichiers se trouvent dans l'artefact d'entrée TemplateSource, obtenu à l'étape Source. L'extrait utilise le mode d'action REPLACE_ON_FAILURE. Si la création de la pile échoue, le pipeline la remplace afin que vous n'ayez pas à nettoyer ou résoudre les problèmes de la pile avant d'exécuter à nouveau le pipeline. Le mode d'action est utile pour l'itération rapide sur les piles de test. Pour la RoleArn propriété, la valeur est un rôle de AWS CloudFormation service déclaré ailleurs dans le modèle.

L'action ApproveTestStack interrompt le pipeline et envoie une notification à l'adresse e-mail que vous avez spécifiée lors de la création de la pile du pipeline. Pendant que le pipeline est suspendu, vous pouvez vérifier la pile de WordPress tests et ses ressources. CodePipeline À utiliser pour [approuver ou rejeter](#) cette action. La propriété CustomData comprend une description de l'action en cours d'approbation que le pipeline ajoute à l'e-mail de notification.

Après avoir approuvé cette action, CodePipeline passe à l>DeleteTestStackaction et supprime la WordPress pile de tests et ses ressources.

```
- Name: TestStage
  Actions:
    - Name: CreateStack
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: REPLACE_ON_FAILURE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref TestStackName
        TemplateConfiguration: !Sub "TemplateSource::${TestStackConfig}"
```

```

    TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
  RunOrder: '1'
- Name: ApproveTestStack
  ActionTypeId:
    Category: Approval
    Owner: AWS
    Provider: Manual
    Version: '1'
  Configuration:
    NotificationArn: !Ref CodePipelineSNSTopic
    CustomData: !Sub 'Do you want to create a change set against the production
stack and delete the ${TestStackName} stack?'
  RunOrder: '2'
- Name: DeleteTestStack
  ActionTypeId:
    Category: Deploy
    Owner: AWS
    Provider: CloudFormation
    Version: '1'
  Configuration:
    ActionMode: DELETE_ONLY
    RoleArn: !GetAtt [CFNRole, Arn]
    StackName: !Ref TestStackName
  RunOrder: '3'

```

Étape 3 : ProdStage

L'étape ProdStage du pipeline crée un jeu de modifications par rapport à la pile de production existante, attend l'approbation et exécute ensuite ce jeu de modifications.

Un ensemble de modifications fournit un aperçu de toutes les modifications AWS CloudFormation qui seront apportées à votre stack de production avant de les implémenter. La pile de production ne sera pas opérationnelle lors de votre première exécution du pipeline. L'ensemble de modifications indique les actions AWS CloudFormation effectuées lors de la création de la pile de tests. Pour créer l'ensemble de modifications, l'CreateChangeSetaction utilise le modèle d' WordPress exemple et la configuration du modèle de production à partir de l'artefact TemplateSource d'entrée.

A l'instar de l'étape précédente, l'action ApproveChangeSet interrompt le pipeline et envoie une notification par e-mail. Lorsque le pipeline est suspendu, vous pouvez consulter l'ensemble de modifications pour vérifier toutes les modifications proposées pour la WordPress pile de production. CodePipeline À utiliser pour [approuver ou rejeter](#) cette action afin de poursuivre ou d'arrêter le pipeline, respectivement.

Une fois que vous avez approuvé cette `ExecuteChangeSet` action, celle-ci exécute l'ensemble de modifications, de sorte que AWS CloudFormation toutes les actions décrites dans le jeu de modifications sont exécutées. Pour le cycle initial, AWS CloudFormation crée la pile WordPress de production. Lors des exécutions suivantes, AWS CloudFormation la met à jour.

```
- Name: ProdStage
  Actions:
    - Name: CreateChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: CHANGE_SET_REPLACE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref ProdStackName
        ChangeSetName: !Ref ChangeSetName
        TemplateConfiguration: !Sub "TemplateSource::${ProdStackConfig}"
        TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
      RunOrder: '1'
    - Name: ApproveChangeSet
      ActionTypeId:
        Category: Approval
        Owner: AWS
        Provider: Manual
        Version: '1'
      Configuration:
        NotificationArn: !Ref CodePipelineSNSTopic
        CustomData: !Sub 'A new change set was created for the ${ProdStackName} stack.
Do you want to implement the changes?'
      RunOrder: '2'
    - Name: ExecuteChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CHANGE_SET_EXECUTE
        ChangeSetName: !Ref ChangeSetName
```

```
RoleArn: !GetAtt [CFNRole, Arn]
StackName: !Ref ProdStackName
RunOrder: '3'
```

Étape 3 : Afficher la WordPress pile

Au CodePipeline cours du pipeline, il est utilisé CloudFormation pour créer des piles de test et de production. Pour voir l'état de ces piles et leur sortie, utilisez la CloudFormation console.

Pour afficher une pile

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation/>.
2. Selon que votre pipeline se trouve à l'étape de test ou de production, choisissez la pile Test - MyWordPressSite ou Prod-MyWordPressSite.
3. Pour vérifier l'état de votre pile, affichez les [événements](#) de la pile.

Si la pile est en état d'échec, affichez la raison de l'état pour trouver l'erreur de la pile. Corrigez l'erreur et exécutez de nouveau le pipeline. Si la pile est en bon CREATE_COMPLETE état, consultez ses sorties pour obtenir l'URL de votre WordPress site.

Vous l'avez utilisé avec succès CodePipeline pour créer un flux de livraison continue pour un exemple de WordPress site. Si vous soumettez des modifications au compartiment S3, détecte CodePipeline automatiquement une nouvelle version, puis réexécute votre pipeline. Ce flux de travail facilite l'envoi et le test de modifications avant d'apporter des changements à votre site de production.

Étape 4 : Nettoyer les ressources

Pour vous assurer que des services inutilisés ne vous sont pas facturés, supprimez vos ressources.

Important

Supprimez les WordPress piles de test et de production avant de supprimer la pile de pipeline. La pile de pipelines contient un rôle de service requis pour supprimer les WordPress piles. Si vous avez d'abord supprimé la pile de pipelines, vous pouvez associer un autre rôle de service Amazon Resource Name (ARN) aux WordPress piles, puis les supprimer.

Pour supprimer des objets dans le magasin d'artefacts

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Choisissez le compartiment S3 CodePipeline utilisé comme magasin d'artefacts pour votre pipeline.

Le nom du compartiment utilise le format suivant : *stackname-artifactstorebucket-id*. Si vous avez suivi cette procédure, le nom du compartiment peut ressembler à l'exemple suivant : `sample-WordPress-pipeline-artifactstorebucket-12345abcd12345`.

3. Supprimez tous les objets dans le compartiment S3 du magasin d'artefacts.

Lorsque vous supprimerez la pile du pipeline à l'étape suivante, ce compartiment doit être vide. Sinon, je CloudFormation ne pourrai pas supprimer le compartiment.

Pour supprimer les piles

1. Dans la CloudFormation console, choisissez la pile que vous souhaitez supprimer.

Si les WordPress piles créées par le pipeline sont toujours en cours d'exécution, choisissez-les d'abord. Par défaut, les noms de pile sont `Test-MyWordPressSite` et `Prod-MyWordPressSite`.

Si vous avez déjà supprimé les WordPress piles, choisissez-les `sample-WordPress-pipeline`.

2. Choisissez Actions, puis Supprimer pile.
3. Dans le message de confirmation, choisissez Oui, supprimer.

CloudFormation supprime toutes les ressources de la pile, telles que l'instance EC2, le sujet de notification, le rôle de service et le pipeline.

Maintenant que vous savez comment créer un CloudFormation flux de travail de base CodePipeline, vous pouvez utiliser le modèle d'exemple et les artefacts comme point de départ pour créer le vôtre.

Consultez aussi

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ces paramètres.

- Pour plus d'informations sur les paramètres CloudFormation d'action dans CodePipeline, consultez la [référence de configuration des actions de CloudFormation déploiement](#) dans le Guide de AWS CodePipeline l'utilisateur.
- Pour des valeurs d'exemple de modèle par fournisseur d'action (par exemple pour les champs Owner ou configuration), veuillez consulter la rubrique [Référence sur la structure des actions](#) dans le Guide de l'utilisateur AWS CodePipeline .
- Pour télécharger des modèles de piles de pipeline en YAML ou JSON, consultez [Tutoriel : Créer un pipeline avec CloudFormation](#) dans le Guide de l'utilisateur AWS CodePipeline .

CloudFormation référence aux propriétés de configuration

Lorsque vous créez un CodePipeline pipeline, vous ajoutez une Deploy action au pipeline en CloudFormation tant que fournisseur. Vous devez ensuite spécifier l' CloudFormation action invoquée par le pipeline et les paramètres de l'action. Cette rubrique décrit les propriétés de configuration CloudFormation . Pour spécifier les propriétés, vous pouvez utiliser la CodePipeline console ou créer un objet JSON à utiliser pour l' AWS CLI CodePipeline API ou les CloudFormation modèles.

Rubriques

- [Propriétés de configuration \(Console\)](#)
- [Propriétés de configuration \(objet JSON\)](#)
- [Consultez aussi](#)

Propriétés de configuration (Console)

La CodePipeline [console](#) affiche les propriétés de configuration et indique les propriétés requises en fonction du mode d'action que vous choisissez.

Note

Lorsque vous créez un nouveau pipeline, vous pouvez spécifier uniquement les modes d'action Créer ou mettre à jour une pile ou Créer ou remplacer un jeu de modifications. Les propriétés de la section Avancé sont disponibles uniquement lorsque vous modifiez un pipeline.

Mode d'action

L' CloudFormation action qui est CodePipeline invoquée lors du traitement de l'étape associée. Choisissez l'un des modes d'action suivants :

- Créer ou remplacer un jeu de modifications permet de créer le jeu de modifications s'il n'existe pas, en fonction du nom de la pile et du modèle que vous soumettez. Si l'ensemble de modifications existe, CloudFormation il est supprimé, puis en crée un nouveau.
- Créer ou mettre à jour une pile crée la pile si la pile spécifiée n'existe pas. Si la pile existe, CloudFormation met à jour la pile. Utilisez cette action pour mettre à jour les piles existantes. CodePipeline ne remplacera pas la pile.
- Supprimer une pile supprime une pile. Si vous spécifiez une pile qui n'existe pas, l'action se termine avec succès sans supprimer de pile.
- Exécuter un jeu de modifications exécute un jeu de modifications.
- Remplacer une pile ayant échoué crée la pile si la pile spécifiée n'existe pas. Si la pile existe et est dans un état défaillant (signalé comme ROLLBACK_COMPLETE, ROLLBACK_FAILED, CREATE_FAILED, DELETE_FAILED, ou UPDATE_ROLLBACK_FAILED), CloudFormation supprime la pile puis en crée une nouvelle. Si la pile n'est pas en état d'échec, mettez-la CloudFormation à jour. Utilisez cette action pour remplacer les piles ayant échoué sans les récupérer ou les dépanner. Généralement, ce mode est choisi pour le test.

Nom de la pile

Nom associé à une pile existante ou à une pile que vous souhaitez créer. Le nom doit être unique dans la AWS région dans laquelle vous créez la pile.

Note

Un nom de pile ne peut contenir que des caractères alphanumériques (sensibles à la casse) et des traits d'union. Il doit commencer par un caractère alphabétique et ne doit pas dépasser 128 caractères.

Modifier le nom du jeu

Nom d'un jeu existant de modifications ou d'un nouveau jeu que vous souhaitez créer pour la pile spécifiée.

Modèle

Emplacement d'un fichier CloudFormation modèle, qui suit le format *ArtifactName::TemplateFileName*.

Configuration de modèle

Emplacement d'un fichier de configuration de modèle conforme au format *ArtifactName::TemplateConfigurationFileName*. Le fichier de configuration de modèle peut contenir des valeurs de paramètres du modèle, une politique de pile et des balises. Si vous incluez des informations sensibles, telles que des mots de passe, limitez l'accès à ce fichier. Pour de plus amples informations, veuillez consulter [CloudFormation artefacts](#).

Capacités

Pour les piles contenant certaines ressources, reconnaissance explicite CloudFormation susceptible de créer ou de mettre à jour ces ressources. Par exemple, vous devez spécifier CAPABILITY_IAM si votre modèle de pile contient des ressources Gestion des identités et des accès AWS (IAM). Pour plus d'informations, consultez les paramètres de requête de l'opération d'API [CreateStack](#).

Si vous disposez de ressources IAM dans votre modèle de pile, vous devez spécifier cette propriété.

Vous pouvez spécifier plusieurs capacités.

Nom de rôle

Nom du rôle de service IAM CloudFormation assumé lorsqu'il agit sur les ressources de la pile spécifiée.

Nom du fichier de sortie

Dans la section Avancé, vous pouvez spécifier un nom de fichier de sortie, tel que `CreateStackOutput.json`, qui s'ajoute à l'[artefact de sortie](#) après l'exécution de l'action spécifiée. L'artefact de sortie contient un fichier JSON contenant le contenu de la Outputs section du CloudFormation modèle.

Si vous ne spécifiez pas de nom, CodePipeline cela ne génère pas d'artefact en sortie.

Remplacements de paramètres

Les paramètres sont définis dans votre modèle et vous permettent d'entrer des valeurs personnalisées lorsque vous créez ou mettez à jour une pile. Vous pouvez spécifier un objet JSON qui remplace les valeurs des paramètres du modèle dans le fichier de configuration du

modèle. Tous les noms de paramètres doivent être présents dans le modèle de la pile. Pour de plus amples informations, veuillez consulter [CloudFormation Parameters syntaxe du modèle](#).

Note

Taille maximum de 1 kilo-octets pour l'objet JSON pouvant être stocké dans la propriété `ParameterOverrides`.

Nous vous recommandons d'utiliser le fichier de configuration du modèle pour spécifier la plupart des valeurs de paramètres. Utilisez les remplacements de paramètres pour spécifier des valeurs de paramètres dynamiques uniquement. Les paramètres dynamiques sont inconnus tant que vous n'exécutez pas le pipeline.

L'exemple suivant définit une valeur pour le paramètre `ParameterName` en utilisant une fonction de remplacement de paramètre. La fonction récupère une valeur à partir d'un artefact `CodePipeline` d'entrée. Pour plus d'informations sur les fonctions de remplacement des paramètres, consultez [Utilisation de fonctions de remplacement de paramètres avec des pipelines CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Propriétés de configuration (objet JSON)

Lorsque vous spécifiez `CloudFormation` comme fournisseur d'une action d'étape, définissez les propriétés suivantes dans la propriété `Configuration`. Utilisez l'objet JSON pour `AWS CLI`, `CodePipelineAPI` ou les `CloudFormation` modèles. Pour obtenir des exemples, consultez [Procédure : Création d'un pipeline pour les piles de test et de production](#) et [CloudFormation référence aux propriétés de configuration](#).

ActionMode

L' `CloudFormation` action qui est `CodePipeline` invoquée lorsqu'elle traite l'étape associée. Spécifiez un seul des modes d'action suivants :

- `CHANGE_SET_EXECUTE` exécute un jeu de modifications.

- **CHANGE_SET_REPLACE** crée le jeu de modification, s'il n'existe pas, en fonction du nom de la pile et du modèle que vous soumettez. Si l'ensemble de modifications existe, CloudFormation il est supprimé, puis en crée un nouveau.
- **CREATE_UPDATE** crée la pile si la pile spécifiée n'existe pas. Si la pile existe, CloudFormation la met à jour. Utilisez cette action pour mettre à jour les piles existantes. CodePipeline ne remplacera pas la pile.
- **DELETE_ONLY** supprime une pile. Si vous spécifiez une pile qui n'existe pas, l'action se termine avec succès sans supprimer de pile.
- **REPLACE_ON_FAILURE** crée une pile si la pile spécifiée n'existe pas. Si la pile existe et est dans un état défaillant (signalé comme **ROLLBACK_COMPLETE**, **ROLLBACK_FAILED**, **CREATE_FAILED**, **DELETE_FAILED**, ou **UPDATE_ROLLBACK_FAILED**), CloudFormation supprime la pile puis en crée une nouvelle. Si la pile n'est pas en état d'échec, mettez-la CloudFormation à jour. Utilisez cette action pour remplacer automatiquement les piles ayant échoué sans les récupérer ou les dépanner. Généralement, ce mode est choisi pour le test.

Cette propriété est requise.

Capabilities

Pour les piles contenant certaines ressources, reconnaissance explicite CloudFormation susceptible de créer ou de mettre à jour ces ressources. Par exemple, vous devez spécifier **CAPABILITY_IAM** si votre modèle de pile contient des ressources Gestion des identités et des accès AWS (IAM). Pour plus d'informations, consultez les paramètres de requête de l'opération d'API [CreateStack](#).

Cette propriété est conditionnelle. Si vous disposez de ressources IAM dans votre modèle de pile, vous devez spécifier cette propriété.

Vous pouvez spécifier plusieurs fonctionnalités. L'exemple suivant ajoute les propriétés **CAPABILITY_IAM** et **CAPABILITY_AUTO_EXPAND** au modèle :

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_IAM, CAPABILITY_AUTO_EXPAND
  ChangeSetName: pipeline-changeset
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
```

```
TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'  
TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {  
  "ActionMode": "CHANGE_SET_REPLACE",  
  "Capabilities": "CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND",  
  "ChangeSetName": "pipeline-changeset",  
  "RoleArn": "CloudFormation_Role_ARN",  
  "StackName": "my-pipeline-stack",  
  "TemplateConfiguration": "my-pipeline-stack::template-  
configuration.json",  
  "TemplatePath": "my-pipeline-stack::template-export.yml"  
}
```

ChangeSetName

Nom d'un jeu existant de modifications ou d'un nouveau jeu que vous souhaitez créer pour la pile spécifiée.

Cette propriété est requise pour les modes d'action suivants : CHANGE_SET_REPLACE et CHANGE_SET_EXECUTE. Pour tous les autres modes d'action, cette propriété est ignorée.

OutputFileName

Nom du fichier de sortie, tel que `CreateStackOutput.json`. CodePipeline ajoute le fichier à l'[artefact de sortie](#) après avoir effectué l'action spécifiée. L'artefact de sortie contient un fichier JSON contenant le contenu de la `Outputs` section du CloudFormation modèle.

Cette propriété est facultative. Si vous ne spécifiez pas de nom, CodePipeline cela ne génère pas d'artefact en sortie.

ParameterOverrides

Les paramètres sont définis dans votre modèle et vous permettent d'entrer des valeurs personnalisées lorsque vous créez ou mettez à jour une pile. Vous pouvez spécifier un objet JSON qui remplace les valeurs des paramètres du modèle dans le fichier de configuration du modèle. Tous les noms de paramètres doivent être présents dans le modèle de la pile. Pour de plus amples informations, veuillez consulter [CloudFormation Parameters syntaxe du modèle](#).

L'exemple suivant ajoute les remplacements de paramètres `InstanceType` et `KeyName` au modèle :

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_NAMED_IAM
  ChangeSetName: pipeline-changeset
  ParameterOverrides: '{"InstanceType": "t2.small","KeyName": "my-keypair"}'
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {
  "ActionMode": "CHANGE_SET_REPLACE",
  "Capabilities": "CAPABILITY_NAMED_IAM",
  "ChangeSetName": "pipeline-changeset",
  "ParameterOverrides": "{\"InstanceType\": \"t2.small\", \"KeyName\": \"my-keypair\"}",
  "RoleArn": "CloudFormation_Role_ARN",
  "StackName": "my-pipeline-stack",
  "TemplateConfiguration": "my-pipeline-stack::template-configuration.json",
  "TemplatePath": "my-pipeline-stack::template-export.yml"
}
```

Note

La taille maximale de l'objet JSON qui peut être stocké dans la propriété `ParameterOverrides` est de 1 kilo-octet.

Nous vous recommandons d'utiliser le fichier de configuration du modèle pour spécifier la plupart des valeurs de paramètres. Utilisez les remplacements de paramètres pour spécifier des valeurs de paramètres dynamiques uniquement. Les valeurs des paramètres dynamiques sont inconnues jusqu'à ce que vous exécutiez le pipeline.

L'exemple suivant définit une valeur pour le paramètre `ParameterName` en utilisant une fonction de remplacement de paramètre. La fonction récupère une valeur à partir d'un artefact `CodePipeline` d'entrée. Pour plus d'informations sur les fonctions de remplacement des

paramètres, consultez [Utilisation de fonctions de remplacement de paramètres avec des pipelines CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Cette propriété est facultative.

RoleArn

Nom de ressource Amazon (ARN) du rôle de service IAM CloudFormation assumé lorsqu'il agit sur les ressources d'une pile.

Cette propriété est obligatoire pour les modes d'action suivants : CREATE_UPDATE, REPLACE_ON_FAILURE, DELETE_ONLY, et CHANGE_SET_REPLACE. RoleArn n'est pas appliqué lors de l'exécution d'un jeu de modifications. Si vous ne l'utilisez pas CodePipeline pour créer l'ensemble de modifications, assurez-vous qu'un rôle est associé à l'ensemble ou à la pile de modifications.

StackName

Nom d'une pile existante ou d'une pile que vous souhaitez créer.

Cette propriété est requise pour tous les modes d'action.

TemplateConfiguration

TemplateConfiguration est le fichier de configuration de modèle. Vous incluez ce fichier dans un artefact d'entrée pour cette action. Le nom de fichier de configuration de modèle suit ce format :

Artifactname::TemplateConfigurationFileName

Artifactname est le nom de l'artefact d'entrée tel qu'il apparaît dans CodePipeline. Par exemple, une étape source avec le nom d'artefact SourceArtifact et un nom de fichier test-configuration.json crée un nom TemplateConfiguration, tel qu'illustré dans l'exemple suivant :

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

Le fichier de configuration de modèle peut contenir des valeurs de paramètres du modèle et une politique de pile. Si vous incluez des informations sensibles, telles que des mots de passe, limitez l'accès à ce fichier. Pour un exemple de fichier de configuration, consultez [CloudFormation artefacts](#).

Cette propriété est facultative.

TemplatePath

TemplatePath représente le fichier CloudFormation modèle. Vous incluez ce fichier dans un artefact d'entrée pour cette action. Le nom de fichier respecte le format suivant :

Artifactname::TemplateFileName

Artifactname est le nom de l'artefact d'entrée tel qu'il apparaît dans CodePipeline. Par exemple, une étape source avec le nom d'artefact SourceArtifact et un nom de fichier template.yaml crée un nom TemplatePath, tel qu'illustré dans l'exemple suivant :

```
"TemplatePath": "SourceArtifact::template.yaml"
```

Cette propriété est requise pour les modes d'action suivants : CREATE_UPDATE, REPLACE_ON_FAILURE et CHANGE_SET_REPLACE. Pour tous les autres modes d'action, cette propriété est ignorée.

Consultez aussi

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ces paramètres.

- Pour plus d'informations sur les paramètres CloudFormation d'action dans CodePipeline, consultez la [référence de configuration des actions de CloudFormation déploiement](#) dans le Guide de AWS CodePipeline l'utilisateur.
- Pour des valeurs d'exemple de modèle par fournisseur d'action (par exemple pour les champs Owner ou configuration), veuillez consulter la rubrique [Référence sur la structure des actions](#) dans le Guide de l'utilisateur AWS CodePipeline .
- Pour télécharger des modèles de piles de pipeline en YAML ou JSON, consultez [Tutoriel : Créer un pipeline avec CloudFormation](#) dans le Guide de l'utilisateur AWS CodePipeline .

CloudFormation artefacts

CodePipeline exécute des tâches sur des artefacts comme CodePipeline exécute un pipeline. En effet CloudFormation, les artefacts peuvent inclure un fichier de modèle de pile, un fichier de configuration de modèle, ou les deux. CodePipeline utilise ces artefacts pour travailler avec des CloudFormation piles et des ensembles de modifications.

Si vous utilisez Amazon Simple Storage Service (Amazon S3) comme référentiel source, vous devez compresser le modèle et les fichiers de configuration du modèle dans un seul fichier avant de les charger dans un compartiment S3. Pour les autres référentiels, tels que GitHub et AWS CodeCommit, téléchargez des artefacts sans les compresser. Pour plus d'informations, consultez [Création d'un pipeline, des étapes et des actions](#) dans le Guide de l'utilisateur AWS CodePipeline .

Vous pouvez ajouter autant de fichiers que nécessaire dans votre référentiel. Par exemple, vous pouvez inclure deux configurations différentes pour le même modèle : une configuration de test et une configuration de production.

Cette rubrique décrit chaque type d'artefact.

Rubriques

- [Fichier de modèle de pile](#)
- [Fichier de configuration de modèle](#)
- [Consultez aussi](#)

Fichier de modèle de pile

Un fichier de modèle de pile définit les ressources que CloudFormation provisionnent et configurent. Ces fichiers sont les mêmes modèles que ceux que vous utilisez lorsque vous créez ou mettez à jour des piles à l'aide CloudFormation de. Vous pouvez utiliser des modèles au format YAML ou JSON. Pour plus d'informations sur les modèles, consultez [CloudFormation sections du modèle](#).

Fichier de configuration de modèle

Un fichier de configuration de modèle est un fichier texte au format JSON qui peut spécifier les valeurs des paramètres du modèle ou une [politique de pile](#) et les balises. Utilisez ces fichiers de configuration pour spécifier les valeurs des paramètres ou une politique de pile pour une pile. Toutes les valeurs des paramètres que vous spécifiez doivent être déclarées dans le modèle associé.

Si vous incluez des informations sensibles (telles que des mots de passe) dans ce fichier, limitez l'accès à ce fichier. Par exemple, si vous chargez votre artefact dans un compartiment S3, utilisez [des politiques de compartiment S3 ou des politiques utilisateur](#) pour en limiter l'accès.

Pour créer un fichier de configuration, utilisez le format suivant :

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

L'exemple suivant spécifie TestEC2Key comme paramètre KeyName, ajoute une balise Department dont la valeur est Marketing et ajoute une politique de pile qui permet toutes les actions de mise à jour, à l'exception d'une mise à jour qui supprimerait une ressource.

```
{
  "Parameters" : {
    "KeyName" : "TestEC2Key"
  },
  "Tags" : {
    "Department" : "Marketing"
  },
  "StackPolicy" : {
    "Statement" : [
      {
        "Effect" : "Allow",
        "NotAction" : "Update:Delete",
        "Principal": "*",
        "Resource" : "*"
      }
    ]
  }
}
```

```
}  
}
```

Consultez aussi

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ces paramètres.

- Pour plus d'informations sur les paramètres CloudFormation d'action dans CodePipeline, consultez la [référence de configuration des actions de CloudFormation déploiement](#) dans le Guide de AWS CodePipeline l'utilisateur.
- Pour des valeurs d'exemple de modèle par fournisseur d'action (par exemple pour les champs `Owner` ou `configuration`), veuillez consulter la rubrique [Référence sur la structure des actions](#) dans le Guide de l'utilisateur AWS CodePipeline .
- Pour télécharger des modèles de piles de pipeline en YAML ou JSON, consultez [Tutoriel : Créer un pipeline avec CloudFormation](#) dans le Guide de l'utilisateur AWS CodePipeline .

Utilisation de fonctions de remplacement de paramètres avec des pipelines CodePipeline

Au cours d'une CodePipeline étape, vous pouvez définir des [remplacements de paramètres](#) pour les CloudFormation actions. Les remplacements de paramètres vous permettent de spécifier des valeurs de paramètres de modèle qui remplacent les valeurs d'un fichier de configuration de modèle. CloudFormation fournit des fonctions qui vous aident à spécifier des valeurs dynamiques (valeurs inconnues tant que le pipeline ne fonctionne pas).

Rubriques

- [Fn::GetArtifactAtt](#)
- [Fn::GetParam](#)
- [Consultez aussi](#)

Fn::GetArtifactAtt

La fonction `Fn::GetArtifactAtt` récupère la valeur d'un attribut à partir d'un artefact d'entrée, comme le nom du compartiment S3 dans lequel l'artefact est stocké. Utilisez cette fonction pour spécifier les attributs d'un artefact, tels que son nom de fichier ou son nom de Amazon S3 compartiment.

Lorsque vous exécutez un pipeline, il CodePipeline copie et écrit des fichiers dans le magasin d'artefacts du pipeline (un compartiment S3). CodePipeline génère les noms de fichiers dans le magasin d'artefacts. Ces noms de fichiers sont inconnus avant l'exécution du pipeline.

Par exemple, dans votre pipeline, vous pouvez avoir un stage source qui CodePipeline copie le code source de votre AWS Lambda fonction dans le magasin d'artefacts. À l'étape suivante, vous disposez d'un CloudFormation modèle qui crée la fonction Lambda, mais CloudFormation nécessite le nom du fichier pour créer la fonction. Vous devez utiliser la fonction `Fn::GetArtifactAtt` pour transmettre les noms exacts des fichiers et du compartiment S3.

Syntaxe

Utilisez la syntaxe suivante pour récupérer la valeur d'attribut d'un artefact.

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

`artifactName`

Nom de l'artefact d'entrée. Vous devez déclarer cet artefact comme entrée pour l'action associée.

`attributeName`

Nom de l'attribut d'artefact dont vous voulez récupérer la valeur. Pour plus d'informations sur chaque attribut d'artefact, consultez la section Attributs qui suit.

Exemple

Les remplacements de paramètres suivants spécifient les paramètres `BucketName` et `ObjectKey` en récupérant le nom du compartiment S3 et le nom de l'artefact `LambdaFunctionSource`. Cet exemple suppose que le code source de la fonction Lambda a été CodePipeline copié et enregistré en tant qu'artefact, par exemple dans le cadre d'une étape source.

```
{  
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"] },  
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"] }  
}
```

Attributs

Vous pouvez récupérer les attributs suivants pour un artefact.

BucketName

Nom du compartiment S3 dans lequel l'artefact est stocké.

ObjectKey

Nom du .zip fichier contenant l'artefact généré par CodePipeline, par exemple1ABCyZZ.zip.

URL

URL Amazon Simple Storage Service (Amazon S3) de l'artefact, comme https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip.

Fn::GetParam

La fonction `Fn::GetParam` renvoie une valeur à partir d'une paire clé-valeur dans un fichier au format JSON. Le fichier JSON doit être inclus dans un artefact.

Utilisez cette fonction pour récupérer les valeurs de sortie d'une CloudFormation pile et les utiliser comme entrée pour une autre action. Par exemple, si vous spécifiez le nom du fichier de sortie pour une CloudFormation action, CodePipeline enregistrez le résultat dans un fichier JSON, puis l'ajoutez au .zip fichier de l'artefact de sortie. Utilisez la fonction `Fn::GetParam` pour récupérer la valeur de sortie et l'utiliser comme entrée pour une autre action.

Syntaxe

Utilisez la syntaxe suivante pour récupérer une valeur à partir d'une paire clé-valeur.

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

artifactName

Nom de l'artefact qui doit être inclus comme artefact d'entrée pour l'action associée.

JSONFileName

Nom d'un fichier JSON contenu dans l'artefact.

keyName

Nom de la clé dont vous voulez récupérer la valeur.

Exemples

Les exemples suivants montrent comment utiliser la fonction `Fn::GetParam` dans un remplacement de paramètre.

Syntaxe

Le remplacement de paramètre suivant spécifie le paramètre `WebSiteURL` en récupérant la valeur de la clé `URL` à partir du fichier `stack-output.json` situé dans l'artefact `WebStackOutput`.

```
{
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"]}
}
```

CloudFormation extraits de modèles

Les extraits de CloudFormation modèle suivants, issus d'un CodePipeline pipeline, montrent comment transmettre des sorties de pile. Ces extraits présentent deux étapes de la définition du pipeline. La première étape crée une pile et enregistre ses sorties dans le fichier `TestOutput.json` dans l'artefact `StackAOutput`. Ces valeurs sont spécifiées par les propriétés `OutputFileName` et `OutputArtifacts`.

Le nom de l'artefact d'entrée source pour les étapes est `TemplateSource`. Le nom de fichier pour le modèle de pile est `teststackA.yaml`, et le nom du fichier de configuration est `test-configuration.json`. Dans les deux étapes, ces valeurs sont spécifiées pour les propriétés `TemplateConfiguration` et `TemplatePath` comme indiqué :

```
TemplateConfiguration: TemplateSource::test-configuration.json
TemplatePath: TemplateSource::teststackA.yaml
```

ExempleÉtape Créer pile A

```
- Name: CreateTestStackA
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
```



```

Configuration:
  ActionMode: CREATE_UPDATE
  Capabilities: CAPABILITY_IAM
  OutputFileName: TestOutput.json
  RoleArn: !GetAtt [CFNRole, Arn]
  StackName: StackA
  TemplateConfiguration: TemplateSource::test-configuration.json
  TemplatePath: TemplateSource::teststackA.yaml
InputArtifacts:
  - Name: TemplateSource
OutputArtifacts:
  - Name: StackAOutput
RunOrder: '1'

```

Dans une étape ultérieure, la pile B utilise les sorties de la pile A. Dans la propriété `ParameterOverrides`, l'exemple utilise la fonction `Fn::GetParam` pour spécifier le paramètre `StackBInputParam`. La valeur obtenue est la valeur associée à la clé `StackAOutputName`.

ExempleÉtape Créer pile B

```

- Name: CreateTestStackB
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackB
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackB.yaml
        ParameterOverrides: |
          {
            "StackBInputParam" : { "Fn::GetParam" : ["StackAOutput", "TestOutput.json",
"StackAOutputName"]}
          }
      InputArtifacts:
        - Name: TemplateSource
        - Name: StackAOutput

```

```
RunOrder: '1'
```

Consultez aussi

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ces paramètres.

- Pour plus d'informations sur les paramètres CloudFormation d'action dans CodePipeline, consultez la [référence de configuration des actions de CloudFormation déploiement](#) dans le Guide de AWS CodePipeline l'utilisateur.
- Pour des valeurs d'exemple de modèle par fournisseur d'action (par exemple pour les champs Owner ou configuration), veuillez consulter la rubrique [Référence sur la structure des actions](#) dans le Guide de l'utilisateur AWS CodePipeline .
- Pour télécharger des modèles de piles de pipeline en YAML ou JSON, consultez [Tutoriel : Créer un pipeline avec CloudFormation](#) dans le Guide de l'utilisateur AWS CodePipeline .

Le guide CloudFormation de référence des modèles

Vous trouverez désormais des informations détaillées sur les ressources et les composants disponibles pour la création de modèles CloudFormation dans le [Guide de référence des modèles CloudFormation](#). Ce guide sert de référence technique pour les modèles et complète le contenu de démarrage du guide de l' CloudFormation utilisateur.

Le Guide de référence des modèles CloudFormation fournit des informations sur :

- Types de ressources et leurs propriétés
- DeletionPolicy et autres attributs au niveau des ressources
- Fonctions intrinsèques, telles que Ref et Sub
- Transformations (CloudFormation macros)
- Scripts d'assistance pour la configuration des EC2 instances
- Définitions des types de ressources lisibles par machine

Ce guide est régulièrement mis à jour pour refléter :

- Nouveaux types de ressources
- Modifications de propriétés
- Nouvelles valeurs renvoyées
- Mises à jour de la documentation

Sécurité dans CloudFormation

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- **Sécurité du cloud** : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à CloudFormation, voir [AWS Services concernés par programme de conformitéAWS](#) .
- **Sécurité dans le cloud** — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation CloudFormation. Les rubriques suivantes expliquent comment procéder à la configuration CloudFormation pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos CloudFormation ressources.

Rubriques

- [CloudFormation Empêcher les piles d'être supprimées](#)
- [Empêcher les mises à jour des ressources de la pile](#)
- [Protection des données dans CloudFormation](#)
- [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#)
- [Journalisation des appels d' CloudFormation API avec AWS CloudTrail](#)
- [Sécurité de l'infrastructure dans CloudFormation](#)
- [Résilience dans CloudFormation](#)
- [Validation de conformité pour CloudFormation](#)

- [Analyse de configuration et de vulnérabilité dans CloudFormation](#)
- [Bonnes pratiques de sécurité pour CloudFormation](#)
- [Accès CloudFormation via un point de terminaison d'interface \(AWS PrivateLink\)](#)

CloudFormation Empêcher les piles d'être supprimées

Vous pouvez empêcher la suppression accidentelle d'une pile en activant la protection contre la résiliation sur la pile. Si un utilisateur tente de supprimer une pile pour laquelle la protection contre la résiliation est activée, la suppression échoue et la pile, et son statut, restent inchangés. Vous pouvez activer la protection contre la résiliation sur une pile lorsque vous la créez. Par défaut, la protection contre la résiliation est désactivée sur les piles. Vous pouvez définir la protection contre la résiliation sur une pile ayant n'importe quel statut sauf `DELETE_IN_PROGRESS` ou `DELETE_COMPLETE`.

L'activation ou la désactivation de la protection pour résilier sur une pile transmet le même choix à toutes les piles imbriquées appartenant à cette pile. Vous ne pouvez pas activer ou désactiver la protection contre la résiliation directement sur une pile imbriquée. Si un utilisateur tente de supprimer directement une pile imbriquée appartenant à une pile pour laquelle la protection contre la résiliation est activée, l'opération échoue et la pile imbriquée reste inchangée.

Toutefois, si un utilisateur effectue une mise à jour d'une pile entraînant la suppression de la pile imbriquée, CloudFormation supprime la pile imbriquée en conséquence.

La protection contre la résiliation n'est pas synonyme de désactivation de la restauration. La protection contre la résiliation s'applique uniquement aux tentatives de suppression des piles, tandis que la désactivation de la restauration s'applique à la restauration automatique lors de l'échec de la création d'une pile.

Pour activer la protection contre la résiliation lors de la création d'une pile

Sur la page Specify stack options (Spécifier les options de pile) de l'assistant Créer une pile, sous Options avancées, développez la section Protection de la résiliation et sélectionnez Activer. Pour de plus amples informations, veuillez consulter [Configuration des options de pile](#).

Pour activer ou désactiver la protection contre la résiliation sur une pile existante

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.

3. Sélectionnez la pile qui vous intéresse.

Note

Si NESTED s'affiche en regard du nom de la pile, cela signifie que la pile est imbriquée. Dans ce cas, vous pouvez seulement modifier la protection contre la résiliation sur la pile racine à laquelle appartient la pile imbriquée.

4. Dans le volet des détails de la pile, sélectionnez Stack actions (Actions de pile), puis Edit termination protection (Modifier la protection de la résiliation).

CloudFormation affiche la boîte de dialogue Modifier la protection contre le licenciement.

5. Choisissez Activer ou Désactiver, puis sélectionnez Enregistrer.

Pour activer ou désactiver la protection contre la résiliation sur une pile imbriquée

Si NESTED s'affiche en regard du nom de la pile, cela signifie que la pile est imbriquée. Dans ce cas, vous pouvez seulement modifier la protection contre la résiliation sur la pile racine à laquelle appartient la pile imbriquée. Pour modifier la protection contre la résiliation sur la racine de la pile :

1. Connectez-vous à la CloudFormation console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation, en haut de l'écran, sélectionnez votre Région AWS.
3. Sélectionnez la pile imbriquée qui vous intéresse.
4. Dans le volet Stack info (Infos de pile), dans la section Présentation, sélectionnez le nom de la pile répertoriée comme Pile racine.

CloudFormation affiche les détails de la pile racine.

5. Choisissez Stack actions (Actions de pile), puis choisissez Edit Termination Protection (Modifier la protection de la résiliation).

CloudFormation affiche la boîte de dialogue Modifier la protection contre le licenciement.

6. Choisissez Activer ou Désactiver, puis sélectionnez Enregistrer.

Pour activer ou désactiver la protection contre la résiliation à l'aide de la ligne de commande.

Utilisez la commande [update-termination-protection](#).

Contrôle des personnes autorisées à modifier la protection contre la résiliation sur les piles

Pour activer ou désactiver la protection contre la résiliation sur des piles, un utilisateur doit être autorisé à effectuer l'action `cloudformation:UpdateTerminationProtection`. Par exemple, la politique ci-dessous permet aux utilisateurs d'activer ou de désactiver la protection contre la résiliation sur des piles.

Pour plus d'informations sur la spécification des autorisations dans CloudFormation, consultez [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

Exemple Exemple de politique qui octroie des autorisations pour modifier la protection contre la résiliation d'une pile

JSON

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

Empêcher les mises à jour des ressources de la pile

Lorsque vous créez une pile, toutes les actions de mise à jour sont autorisées au niveau de toutes les ressources. Par défaut, toute personne disposant des autorisations de mise à jour de la pile peuvent mettre à jour toutes les ressources correspondantes. Au cours d'une mise à jour, certaines ressources peuvent nécessiter une interruption ou être complètement remplacées, ce qui se traduit par un nouvel espace de stockage physique IDs ou un stockage complètement nouveau. Vous pouvez empêcher les ressources de la pile d'être mises à jour ou supprimées par erreur pendant une mise à jour de la pile. Une politique de pile est un document JSON qui définit les actions de mise à jour qui peuvent être effectuées au niveau des ressources désignées.

Une fois que vous avez défini une politique de pile, toutes les ressources de cette dernière sont protégées par défaut. Pour autoriser les mises à jour de ressources spécifiques, vous spécifiez une instruction `Allow` explicite pour ces ressources dans votre politique de pile. Vous ne pouvez définir qu'une seule politique de pile par pile, mais, vous pouvez protéger plusieurs ressources au sein d'une politique unique. Une politique de pile s'applique à tous les utilisateurs CloudFormation qui tentent de mettre à jour la pile. Vous ne pouvez pas associer des politiques de pile différentes à des utilisateurs distincts.

Une politique de pile s'applique uniquement au cours des mises à jour de la pile. Elle ne fournit pas de contrôles d'accès comme une politique Gestion des identités et des accès AWS (IAM). Utilisez uniquement une politique de pile comme mécanisme de sécurité afin d'empêcher les mises à jour accidentelles de ressources spécifiques de la pile. Pour contrôler l'accès aux AWS ressources ou aux actions, utilisez IAM.

Rubriques

- [Exemple de politique de pile](#)
- [Définition d'une politique de pile](#)
- [Définition d'une politique de pile](#)
- [Mise à jour des ressources protégées](#)
- [Modification d'une politique de pile](#)
- [Autres exemples de politiques de pile](#)

Exemple de politique de pile

L'exemple suivant de politique de pile empêche les mises à jour de la `ProductionDatabase` ressource :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
```



```
    "Principal": "*",
    "Resource" : "LogicalResourceId/ProductionDatabase"
  }
]
```

Lorsque vous définissez une politique de pile, toutes les ressources sont protégées par défaut. Pour autoriser les mises à jour de toutes les ressources, nous ajoutons une instruction `Allow`, qui autorise toutes les actions sur toutes les ressources. Bien que l'instruction `Allow` spécifie toutes les ressources, l'instruction `Deny` explicite prévaut pour la ressource associée à l'ID logique `ProductionDatabase`. Cette instruction `Deny` empêche toutes les actions de mise à jour, telles que le remplacement ou la suppression, u niveau de la ressource `ProductionDatabase`.

L'élément `Principal` est obligatoire, mais accepte uniquement le caractère générique (*). Autrement dit, l'instruction s'applique à tous les [mandataires](#).

Note

Lors d'une mise à jour de la pile, met CloudFormation automatiquement à jour les ressources qui dépendent d'autres ressources mises à jour. Par exemple, CloudFormation met à jour une ressource qui fait référence à une ressource mise à jour. CloudFormation n'apporte aucune modification physique, telle que l'ID de la ressource, aux ressources mises à jour automatiquement, mais si une politique de pile est associée à ces ressources, vous devez être autorisé à les mettre à jour.

Définition d'une politique de pile

Lorsque vous créez une pile, aucune politique de pile n'est définie. Dès lors, toutes les actions de mise à jour sont autorisées sur toutes les ressources. Pour protéger les ressources de la pile contre les actions de mise à jour, spécifiez une politique de pile correspondante. Une politique de pile est un document JSON qui définit les actions de mise à jour de CloudFormation pile que CloudFormation les utilisateurs peuvent effectuer et les ressources auxquelles ces actions s'appliquent. Vous définissez la politique de la pile lorsque vous créez cette dernière, en spécifiant un fichier texte qui contient votre politique de pile ou en saisissant les informations requises. Lorsque vous définissez une politique au niveau de votre pile, toute mise à jour qui n'est pas explicitement autorisée est refusée par défaut.

Vous pouvez spécifier une politique de pile avec cinq éléments : `Effect`, `Action`, `Principal`, `Resource` et `Condition`. Le code fictif suivant illustre la syntaxe de la politique de pile.

```
{
  "Statement" : [
    {
      "Effect" : "Deny_or_Allow",
      "Action" : "update_actions",
      "Principal" : "*",
      "Resource" : "LogicalResourceId/resource_logical_ID",
      "Condition" : {
        "StringEquals_or_StringLike" : {
          "ResourceType" : [resource_type, ...]
        }
      }
    }
  ]
}
```

Effect

Détermine si les actions que vous spécifiez sont refusées ou autorisées au niveau des ressources que vous spécifiez. Vous pouvez spécifier uniquement Deny ou Allow (, par exemple).

```
"Effect" : "Deny"
```

Important

Si une politique de pile inclut des déclarations contradictoires (qui autorisent et refusent en même temps les mises à jour d'une ressource), une instruction Deny prévaut toujours sur une instruction Allow. Pour vous assurer qu'une ressource est protégée, utilisez une instruction Deny pour celle-ci.

Action

Spécifie les actions de mise à jour qui sont refusées ou autorisées :

Update:Modify

Spécifie les actions de mise à jour au cours desquelles les ressources peuvent rencontrer quelques interruptions ou aucune lorsque les modifications sont appliquées. Toutes les ressources conservent leur aspect physique IDs.

Update:Replace

Spécifie les actions de mise à jour au cours desquelles les ressources sont recréées. CloudFormation crée une autre ressource avec les mises à jour définies, puis supprime l'ancienne ressource. Etant donné que la ressource est recréée, l'ID physique de la nouvelle ressource peut être différent.

Update>Delete

Spécifie les actions de mise à jour au cours desquelles les ressources sont supprimées. Les mises à jour qui suppriment totalement les ressources d'un modèle de pile requièrent cette action.

Mise à jour:*

Spécifie toutes les actions de mise à jour. L'astérisque est un caractère générique qui représente toutes les actions de mise à jour.

L'exemple suivant montre comment définir uniquement les actions de remplacement et de suppression :

```
"Action" : ["Update:Replace", "Update>Delete"]
```

Pour autoriser toutes les actions de mise à jour à l'exception de l'une d'elles, utilisez `NotAction`. Par exemple, pour autoriser toutes les actions de mise à jour à l'exception de l'action `Update>Delete`, utilisez `NotAction`, comme illustré dans cet exemple :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update>Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Principal

L'élément `Principal` spécifie l'entité à laquelle la politique s'applique. Cet élément est obligatoire, mais accepte uniquement le caractère générique (*). Autrement dit, l'instruction s'applique à tous les [mandataires](#).

Ressource

Spécifie la logique IDs des ressources auxquelles la politique s'applique. Pour spécifier les types de ressources, utilisez l'élément `Condition`.

Pour spécifier une seule ressource, utilisez son ID logique. Par exemple :

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

Vous pouvez utiliser un joker avec logique IDs. Par exemple, si vous utilisez un préfixe d'ID logique commun pour toutes les ressources connexes, vous pouvez tous les spécifier avec un caractère générique :

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

Vous pouvez également utiliser un élément `Not` avec les ressources. Par exemple, pour autoriser les mises à jour de toutes les ressources à l'exception d'une seule, utilisez l'élément `NotResource` afin de protéger cette ressource :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Lorsque vous définissez une politique de pile, toute mise à jour qui n'est pas explicitement autorisée est refusée. En autorisant les mises à jour de toutes les ressources à l'exception de la ressource `ProductionDatabase`, vous refusez les mises à jour de `ProductionDatabase`.

Conditions

Spécifie le type de ressource auquel la politique s'applique. Pour définir la logique IDs de ressources spécifiques, utilisez l'élément `Resource`.

Vous pouvez spécifier un type de ressource, tel que des instances DB RDS et EC2, comme illustré dans l'exemple suivant :

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Principal" : "*",
      "Action" : "Update:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Principal" : "*",
      "Action" : "Update:*",
      "Resource" : "*"
    }
  ]
}
```

L'instruction Allow accorde les autorisations de mise à jour à toutes les ressources et l'instruction Deny refuse les mises à jour des instances DB RDS et EC2. L'instruction Deny prévaut toujours sur les actions autorisées.

Vous pouvez utiliser un caractère générique avec les types de ressource. Par exemple, vous pouvez refuser les autorisations de mise à jour pour toutes les ressources Amazon EC2, telles que les instances, les groupes de sécurité et les sous-réseaux, en utilisant un caractère générique, comme illustré dans l'exemple suivant :

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

Vous devez utiliser la condition `StringLike` lorsque vous utilisez des caractères génériques.

Définition d'une politique de pile

Vous pouvez utiliser la console ou AWS CLI appliquer une politique de pile lorsque vous créez une pile. Vous pouvez également utiliser le AWS CLI pour appliquer une politique de pile à une pile existante. Une fois que vous avez appliqué une politique de pile, vous ne pouvez pas la supprimer de la pile, mais vous pouvez l'utiliser AWS CLI pour la modifier.

Les politiques de pile s'appliquent à tous les CloudFormation utilisateurs qui tentent de mettre à jour la pile. Vous ne pouvez pas associer des politiques de pile différentes à des utilisateurs distincts.

Pour plus d'informations sur la création de politiques de pile, consultez [Définition d'une politique de pile](#).

Pour définir une politique de pile lors de la création de la pile (console)

1. Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.
2. Dans la barre de navigation en haut de l'écran, choisissez le Région AWS pour créer le stack in.
3. Sur la page CloudFormation Stacks, choisissez Create stack.
4. Dans l'assistant Créer une pile, à la page Configure stack options (Configurer les options de pile), développez la section Avancé, puis choisissez Politique de pile.
5. Spécifiez la politique de pile :
 - Pour écrire une politique directement dans la console, choisissez Enter stack policy (Entrer la politique de pile), puis tapez la politique de pile directement dans le champ de texte.
 - Pour utiliser une politique définie dans un fichier distinct, choisissez Upload a file (Charger un fichier) puis Choose file (Choisir un fichier) pour sélectionner le fichier contenant la politique de pile.

Pour définir une politique de pile lors de la création de la pile (AWS CLI)

- Utilisez la commande [create-stack](#) avec l'option `--stack-policy-body` permettant de saisir une stratégie modifiée ou l'option `--stack-policy-url` permettant de spécifier un fichier contenant la stratégie.

Pour définir une politique de pile sur une pile existante (AWS CLI uniquement)

- Utilisez la commande [set-stack-policy](#) avec l'option `--stack-policy-body` pour saisir une politique modifiée, ou l'option `--stack-policy-url` pour spécifier un fichier contenant la politique.

 Note

Pour ajouter une politique à une pile existante, vous devez être autorisé à effectuer cette CloudFormation [SetStackPolicy](#) action.

Mise à jour des ressources protégées

Pour mettre à jour les ressources protégées, créez une politique temporaire qui remplace la politique de la pile et qui autorise les mises à jour de ces ressources. Spécifiez la politique de remplacement lorsque vous mettez à jour la pile. Cette politique ne modifie pas la politique de pile de manière permanente.

Pour mettre à jour les ressources protégées, vous devez être autorisé à utiliser cette CloudFormation [SetStackPolicy](#) action. Pour obtenir des informations sur la définition des autorisations CloudFormation, consultez [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

 Note

Lors d'une mise à jour de la pile, met CloudFormation automatiquement à jour les ressources qui dépendent d'autres ressources mises à jour. Par exemple, CloudFormation met à jour une ressource qui fait référence à une ressource mise à jour. CloudFormation n'apporte aucune modification physique, telle que l'identifiant des ressources, aux ressources mises à jour automatiquement, mais si une politique de pile est associée à ces ressources, vous devez être autorisé à les mettre à jour.

Pour mettre à jour une ressource protégée (console)

- Ouvrez la CloudFormation console à l'adresse <https://console.aws.amazon.com/cloudformation>.

2. Sélectionnez la pile que vous souhaitez mettre à jour, choisissez Stack actions (Actions de pile) puis Mettre à jour la pile.
3. Si vous n'avez pas modifié le modèle de pile, sélectionnez Utiliser modèle en cours, puis cliquez sur Suivant. Si vous avez modifié le modèle, sélectionnez Replace current template (Remplacer le modèle en cours) et spécifiez l'emplacement du modèle mis à jour dans la section Specify template (Spécifier le modèle) :
 - Pour un modèle stocké localement sur votre ordinateur, sélectionnez Upload a template file (Charger un fichier de modèle). Choisissez Choose File (Choisir un fichier) pour accéder au fichier, sélectionnez-le, puis cliquez sur Suivant.
 - Pour un modèle stocké dans un compartiment Amazon S3, sélectionnez URL Amazon S3. Saisissez ou collez l'URL du modèle et cliquez sur Suivant.

Si votre modèle se trouve dans un compartiment dans lequel la gestion des versions est activée, vous pouvez spécifier une version particulière du modèle en ajoutant ?
versionId=*version-id* à l'URL. Pour plus d'informations, consultez [Travail avec des objets dans un compartiment avec gestion des versions activée](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

4. Si votre modèle contient des paramètres, sur la page Specify stack details (Spécifier les détails de pile), entrez ou modifiez les valeurs de paramètre, puis choisissez Suivant.

CloudFormation remplit chaque paramètre avec la valeur actuellement définie dans la pile, à l'exception des paramètres déclarés avec l'NoEchoattribut. Pour utiliser les valeurs actuelles de ces paramètres, choisissez Utiliser valeur existante.

Pour plus d'informations sur l'utilisation de NoEcho pour masquer les informations sensibles, ainsi que sur l'utilisation de paramètres dynamiques pour gérer les secrets, consultez la bonne pratique [N'incorporez pas d'informations d'identification dans vos modèles](#).

5. Spécifiez une politique de pile de remplacement.
 - a. Sur la page Configure stack options (Configurer les options de pile), dans la section Options avancées, sélectionnez Politique de pile.
 - b. Sélectionnez Upload a file (Charger un fichier).
 - c. Cliquez sur Choose file (Choisir un fichier) et accédez au fichier contenant la politique de pile de remplacement ou tapez une politique.
 - d. Choisissez Suivant.

La politique de remplacement doit spécifier une instruction `Allow` pour les ressources protégées que vous souhaitez mettre à jour. Par exemple, pour mettre à jour toutes les ressources protégées, définissez une politique de remplacement temporaire qui permet toutes les mises à jour :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal" : "*",
      "Resource" : "*"
    }
  ]
}
```

 Note

CloudFormation applique la politique de dérogation uniquement lors de cette mise à jour. Cette politique ne modifie pas la politique de pile de manière permanente. Pour modifier une politique de pile, consultez la page [Modification d'une politique de pile](#).

6. Passez en revue les informations de la pile et les modifications que vous avez soumises.

Vérifiez que vous avez soumis les informations correctes, telles que l'URL du modèle ou les valeurs de paramètres appropriées. Si votre modèle contient des ressources IAM, choisissez I acknowledge that this template may create IAM resources (Je sais que ce modèle peut créer des ressources IAM) pour confirmer que vous souhaitez utiliser des ressources IAM dans le modèle. Pour de plus amples informations, veuillez consulter [Reconnaissance des ressources IAM dans les modèles CloudFormation](#).

Dans la section Aperçu de vos modifications, vérifiez que toutes les modifications attendues CloudFormation seront apportées. Par exemple, vérifiez que CloudFormation les ressources que vous vouliez ajouter, supprimer ou modifier sont ajoutées, supprimées ou modifiées. CloudFormation génère cet aperçu en créant un ensemble de modifications pour la pile. Pour de plus amples informations, veuillez consulter [Mettre à jour CloudFormation les piles à l'aide d'ensembles de modifications](#).

7. Lorsque vous êtes satisfait de vos modifications, cliquez sur Update (Mettre à jour).

 Note

À ce stade, vous avez aussi la possibilité d'afficher le jeu de modifications pour vérifier de façon plus complète vos mises à jour proposées. Pour ce faire, cliquez sur Afficher l'ensemble de modifications au lieu de Mettre à jour. CloudFormation affiche l'ensemble de modifications généré en fonction de vos mises à jour. Lorsque vous êtes prêt à effectuer la mise à jour de la pile, cliquez sur Exécuter.

CloudFormation affiche la page des détails de la pile correspondant à votre pile. Votre pile a désormais le statut `UPDATE_IN_PROGRESS`. Une CloudFormation fois la mise à jour de la pile terminée avec succès, l'état de la pile est défini sur `UPDATE_COMPLETE`.

Si la mise à jour de la pile échoue, CloudFormation ; annule automatiquement les modifications et définit le statut de la pile sur `UPDATE_ROLLBACK_COMPLETE`.

Pour mettre à jour une ressource protégée (AWS CLI)

- Utilisez la commande [update-stack](#) avec l'option `--stack-policy-during-update-body` permettant de saisir une stratégie modifiée ou l'option `--stack-policy-during-update-url` permettant de spécifier un fichier contenant la stratégie.

 Note

CloudFormation applique la politique de dérogation uniquement lors de cette mise à jour. Cette politique ne modifie pas la politique de pile de manière permanente. Pour modifier une politique de pile, consultez la page [Modification d'une politique de pile](#).

Modification d'une politique de pile

Pour protéger des ressources supplémentaires ou pour supprimer la protection de ressources, modifiez la politique de pile. Par exemple, lorsque vous ajoutez à votre pile une base de données que vous souhaitez protéger, appliquez une instruction Deny pour cette base de données à la politique de pile. Pour modifier la politique, vous devez avoir l'autorisation d'utiliser l'action [SetStackPolicy](#).

Utilisez le AWS CLI pour modifier les politiques de pile.

Pour modifier une politique de pile (AWS CLI)

- Utilisez la commande [set-stack-policy](#) avec l'option `--stack-policy-body` permettant de saisir une stratégie modifiée ou l'option `--stack-policy-url` permettant de spécifier un fichier contenant la stratégie.

Vous ne pouvez pas supprimer une politique de pile. Pour supprimer toute la protection de toutes les ressources, vous devez modifier la politique pour autoriser explicitement toutes les actions pour toutes les ressources. La politique suivante autorise toutes les mises à jour au niveau de toutes les ressources :

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Autres exemples de politiques de pile

Les exemples suivants de stratégie de pile montre comment empêcher les mises à jour de toutes les ressources de la pile et de ressources spécifiques, ainsi que comment empêcher certains types de mises à jour.

Empêcher les mises à jour de toutes les ressources de la pile

Pour éviter les mises à jour de toutes les ressources d'une pile, la politique suivante spécifie une instruction Deny pour toutes les actions de mise à jour pour toutes les ressources.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
```

```
    "Resource" : "*"
  }
]
}
```

Empêcher les mises à jour d'une seule ressource

La politique suivante refuse toutes les actions de mise à jour de la base de données associée à l'ID logique MyDatabase. Elle autorise toutes les actions de mise à jour au niveau de toutes les autres ressources de la pile avec une instruction Allow. L'instruction Allow ne s'applique pas à la ressource MyDatabase, car l'instruction Deny prévaut toujours sur les actions autorisées.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Vous pouvez obtenir le même résultat que l'exemple précédent en utilisant un refus par défaut. Lorsque vous définissez une politique de stack, elle CloudFormation refuse toute mise à jour qui n'est pas explicitement autorisée. La politique suivante autorise les mises à jour de toutes les ressources, à l'exception de la ressource ProductionDatabase, qui est refusée par défaut.

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

```
]
}
```

Important

L'utilisation d'un refus par défaut présente des risques. Si vous utilisez une instruction Allow ailleurs dans la politique (par exemple, une instruction Allow qui utilise un caractère générique), vous pouvez sans le savoir accorder l'autorisation de mise à jour à des ressources qui ne devraient pas avoir cette autorisation. Comme un refus explicite prévaut sur toutes les actions autorisées, une ressource est réellement protégée avec une instruction Deny.

Empêcher les mises à jour de toutes les Instances d'un type de ressource

La politique suivante refuse que toutes les actions de mise à jour au niveau du type de ressource des instances DB RDS. Elle autorise toutes les actions de mise à jour au niveau de toutes les autres ressources de la pile avec une instruction Allow. L'instruction Allow ne s'applique pas aux ressources des instances DB RDS, car une instruction Deny prévaut toujours sur les actions autorisées.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}
```

Empêcher les mises à jour de remplacement d'une instance

La politique suivante refuse les mises à jour qui entraîneraient un remplacement de l'instance associée à l'ID logique MyInstance. Elle autorise toutes les actions de mise à jour au niveau de toutes les autres ressources de la pile avec une instruction Allow. L'instruction Allow ne s'applique pas à la ressource MyInstance, car l'instruction Deny prévaut toujours sur les actions autorisées.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Empêcher les mises à jour des piles imbriquées

La politique suivante refuse toutes les actions de mise à jour sur le type de ressource de CloudFormation pile (piles imbriquées). Elle autorise toutes les actions de mise à jour au niveau de toutes les autres ressources de la pile avec une instruction Allow. L'instruction Allow ne s'applique pas aux ressources des piles CloudFormation, car une instruction Deny prévaut toujours sur les actions d'autorisation.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",

```

```
    "Condition" : {
      "StringEquals" : {
        "ResourceType" : ["AWS::CloudFormation::Stack"]
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal" : "*",
      "Resource" : "*"
    }
  ]
}
```

Protection des données dans CloudFormation

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS CloudFormation. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée d'AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.

- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez [Norme FIPS \(Federal Information Processing Standard\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec CloudFormation ou d'autres Services AWS utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement au repos

Selon le modèle de responsabilité AWS partagée, CloudFormation stocke vos données cryptées au repos. Il incombe aux clients de définir des politiques de chiffrement et de stockage pour les données stockées dans leurs comptes. Par exemple, nous vous recommandons d'activer le chiffrement au repos pour les modèles et autres données stockées dans des compartiments S3 ou des rubriques SNS. Les clients définissent également les paramètres de chiffrement pour tous les systèmes de stockage de données alloués par CloudFormation.

Chiffrement en transit

CloudFormation utilise des canaux cryptés pour les communications de service dans le cadre du modèle de responsabilité partagée.

Confidentialité du trafic inter-réseau

CloudFormation les communications de service sont cryptées de manière sécurisée par défaut entre les régions ou les zones de disponibilité.

Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS

Avec Gestion des identités et des accès AWS (IAM), vous pouvez créer des utilisateurs IAM et contrôler leur accès à des ressources spécifiques de votre. Compte AWS Lorsque vous utilisez IAM, vous pouvez contrôler ce que les utilisateurs peuvent utiliser CloudFormation, par exemple s'ils peuvent afficher des modèles de pile, créer des piles ou supprimer des piles.

Au-delà CloudFormation des actions spécifiques, vous pouvez gérer les AWS services et les ressources disponibles pour chaque utilisateur. Ainsi, vous pouvez contrôler les ressources auxquelles les utilisateurs peuvent accéder lorsqu'ils les utilisent CloudFormation. Par exemple, vous pouvez spécifier quels utilisateurs peuvent créer des instances Amazon EC2, mettre fin à des instances de base de données ou effectuer des mises à jour. VPCs Ces mêmes autorisations s'appliquent chaque fois qu'ils CloudFormation les utilisent pour effectuer ces actions.

Utilisez les informations des sections suivantes pour contrôler les personnes autorisées à y accéder CloudFormation. Nous verrons également comment autoriser la création de ressources IAM dans des modèles, donner aux applications exécutées sur des instances EC2 les autorisations dont elles ont besoin et utiliser des informations d'identification de sécurité temporaires pour renforcer la sécurité de votre AWS environnement.

Définition de politiques basées sur l'identité IAM pour CloudFormation

Pour donner accès à CloudFormation, vous devez créer et attribuer des politiques IAM qui donnent à vos identités IAM (telles que les utilisateurs ou les rôles) l'autorisation d'appeler les actions d'API dont elles ont besoin.

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. CloudFormation prend en charge des actions, des ressources et des clés de condition spécifiques.

Si vous débutez avec IAM, commencez par vous familiariser avec les éléments d'une politique JSON IAM. Pour plus d'informations, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM. Pour apprendre à créer des politiques IAM, suivez le tutoriel [Créer et associer votre première politique gérée par le client](#) dans la documentation IAM.

Rubriques

- [Actions politiques pour CloudFormation](#)
- [Actions spécifiques à la console pour CloudFormation](#)
- [Ressources politiques pour CloudFormation](#)
- [Clés de conditions de politique pour CloudFormation](#)

Actions politiques pour CloudFormation

Dans l'Action élément de votre déclaration de politique IAM, vous pouvez spécifier toute action d'API proposée CloudFormation . Vous devez préfixer le nom de l'action par la chaîne en minuscules `cloudformation:`. Par exemple : `cloudformation:CreateStack`, `cloudformation:CreateChangeSet` et `cloudformation:UpdateStack`.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules, comme suit :

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques. Par exemple, vous pouvez spécifier toutes les actions dont le nom commence par `Get`, comme suit :

```
"Action": "cloudformation:Get*"
```

Pour consulter la liste complète des actions associées au préfixe de service `cloudformation`, consultez [Actions, ressources et clés de condition pour CloudFormation](#) et [Actions, ressources et clés de condition pour API de commande du Cloud AWS](#) dans la Référence d'autorisation de service.

Exemples

Vous trouverez ci-dessous un exemple de politique d'autorisation qui accorde des autorisations pour consulter les CloudFormation piles.

Exemple 1 : exemple de politique accordant des autorisations d'affichage des piles

JSON

```
{
```

```
"Version":"2012-10-17",
"Statement":[{"Effect":"Allow",
  "Action":["cloudformation:DescribeStacks",
    "cloudformation:DescribeStackEvents",
    "cloudformation:DescribeStackResource",
    "cloudformation:DescribeStackResources"
  ],
  "Resource":""
}]
}
```

Les utilisateurs qui créent ou suppriment des piles ont besoin d'autorisations supplémentaires en fonction de leurs modèles de pile. Par exemple, si votre modèle décrit une file d'attente Amazon SQS, les utilisateurs doivent être autorisés à la fois pour les actions Amazon SQS CloudFormation et pour les actions Amazon SQS, comme indiqué dans l'exemple de politique suivant.

Exemple 2 : exemple de politique accordant les actions de création et d'affichage des piles, ainsi que toutes les actions Amazon SQS

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[{"Effect":"Allow",
    "Action":["sqs:*",
      "cloudformation:CreateStack",
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResources",
      "cloudformation:GetTemplate",
      "cloudformation:ValidateTemplate"
    ],
    "Resource":""
  }]
}
```

Actions spécifiques à la console pour CloudFormation

Les utilisateurs de la CloudFormation console ont besoin d'autorisations supplémentaires en plus de celles requises pour le AWS Command Line Interface ou CloudFormation APIs. Ces autorisations supplémentaires prennent en charge les fonctionnalités spécifiques à la console, telles que le téléchargement de modèles vers des compartiments Amazon S3 et les listes déroulantes pour AWS les types de paramètres spécifiques.

Pour toutes les actions répertoriées ci-dessous, accordez les autorisations sur toutes les ressources ; ne les limitez pas à des piles ou compartiments spécifiques.

L'action suivante est utilisée uniquement par la CloudFormation console et n'est pas documentée dans la référence de l'API. Cette action permet aux utilisateurs d'importer des modèles dans des compartiments Amazon S3.

- `cloudformation:CreateUploadBucket`

Lorsque les utilisateurs chargent des modèles, les utilisateurs ont également besoin des autorisations Amazon S3 suivantes :

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

Pour voir les valeurs dans les listes déroulantes de paramètres pour les modèles avec des types de paramètres AWS spécifiques, les utilisateurs doivent être autorisés à effectuer les appels d'API de description correspondants. Par exemple, les autorisations suivantes sont requises lorsque ces types de paramètres sont utilisés dans le modèle :

- `ec2:DescribeKeyPairs` : obligatoire pour le type de paramètre `AWS::EC2::KeyPair::KeyName`.
- `ec2:DescribeSecurityGroups` : obligatoire pour le type de paramètre `AWS::EC2::SecurityGroup::Id`.
- `ec2:DescribeSubnets` : obligatoire pour le type de paramètre `AWS::EC2::Subnet::Id`.
- `ec2:DescribeVpcs` : obligatoire pour le type de paramètre `AWS::EC2::VPC::Id`.

Pour plus d'informations sur les types de paramètres AWS spécifiques, consultez [Spécifiez les ressources existantes au moment de l'exécution avec les CloudFormation types de paramètres fournis](#).

Ressources politiques pour CloudFormation

Dans une instruction de politique IAM, l'élément `Resource` spécifie l'objet ou les objets couverts par l'instruction. En effet CloudFormation, chaque déclaration de politique IAM s'applique aux ressources que vous spécifiez à l'aide de leur Amazon Resource Names (ARNs). Le format ARN spécifique dépend de la ressource.

Pour une liste complète des types de CloudFormation ressources et de leurs caractéristiques ARNs, voir [Types de ressources définis par CloudFormation](#) dans la référence d'autorisation de service. Pour savoir quelles actions peuvent être associées à l'ARN de chaque ressource, consultez [Actions définies par CloudFormation](#).

Vous pouvez spécifier des actions pour une pile particulière, comme dans l'exemple de politique suivant. Lorsque vous fournissez un ARN, remplacez le champ *placeholder text* par les informations propres à votre ressource.

Exemple 1 : exemple de politique refusant les actions de suppression et de mise à jour pour la pile spécifiée

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloudformation:DeleteStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/MyProductionStack/*"
    }
  ]
}
```

La politique ci-dessus utilise un caractère générique à la fin du nom de la pile, de manière à refuser les actions de suppression et de mise à jour à la fois sur l'ID complet de la pile (comme `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) et sur le nom de la pile (comme `MyProductionStack`).

Pour autoriser des transformations AWS::Serverless à créer un jeu de modifications, incluez l'autorisation de ressource `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31`, comme montré dans l'exemple suivant.

Exemple 2 : exemple de politique autorisant l'action de création d'un jeu de modifications pour la transformation spécifiée

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/Serverless-2016-10-31"
    }
  ]
}
```

Clés de conditions de politique pour CloudFormation

Dans une instruction de politique IAM, vous pouvez définir des conditions facultatives permettant de contrôler quand une politique s'applique. Par exemple, vous pouvez définir une stratégie qui permet aux utilisateurs de créer une pile uniquement lorsqu'ils spécifient une URL de modèle spécifique. Vous pouvez définir CloudFormation des conditions spécifiques et des conditions AWS générales `DateLessThan`, telles que celles qui indiquent à quel moment une politique cesse de prendre effet. Pour plus d'informations et une liste des AWS conditions générales, voir la [référence relative à la condition dans les éléments de stratégie IAM](#) dans le guide de l'utilisateur IAM.

 Note

N'utilisez pas la condition `aws:SourceIp` AWS « wide ». CloudFormation approvisionne les ressources en utilisant sa propre adresse IP, et non l'adresse IP de la demande d'origine. Par exemple, lorsque vous créez une pile, vous envoyez CloudFormation des demandes à partir de son adresse IP pour lancer une instance Amazon EC2 ou pour créer un compartiment Amazon S3, et non à partir de l'adresse IP de l'`CreateStack` appel ou de la commande `create-stack`.

La liste suivante décrit les conditions CloudFormation spécifiques. Ces conditions sont appliquées uniquement lorsque les utilisateurs créent ou mettent à jour les piles :

`cloudformation:ChangeSetName`

Nom du groupe de CloudFormation modifications que vous souhaitez associer à une politique. Utilisez cette condition pour contrôler les jeux de modifications que les utilisateurs peuvent exécuter ou supprimer.

`cloudformation:ImportResourceTypes`

Types de ressource du modèle que vous souhaitez associer à une politique, comme `AWS::EC2::Instance`. Utilisez cette condition pour contrôler les types de ressources que les utilisateurs peuvent utiliser lorsqu'ils importent des ressources dans une pile. Cette condition est vérifiée par rapport aux types de ressources déclarés par les utilisateurs dans le `ResourcesToImport` paramètre, qui n'est actuellement pris en charge que pour AWS CLI les demandes d'API. Lorsque vous utilisez ce paramètre, vous devez spécifier tous les types de ressources que les utilisateurs doivent contrôler pendant les opérations d'importation. Pour plus d'informations sur le paramètre `ResourcesToImport`, consultez l'action [CreateChangeSet](#) dans la Référence API AWS CloudFormation .

Pour une liste des valeurs possibles de `ResourcesToImport`, consultez [Prise en charge des types de ressources](#).

Utilisez la convention de dénomination des ressources en trois parties pour spécifier les types de ressources que les utilisateurs peuvent utiliser, depuis toutes les ressources d'une organisation jusqu'à un type de ressource individuel.

`organization::*`

Spécifiez tous les types de ressources pour une organisation donnée.

*organization::service_name::**

Spécifiez tous les types de ressources pour le service spécifié au sein d'une organisation donnée.

organization::service_name::resource_type

Spécifiez un type de ressource spécifique.

Par exemple :

*AWS::**

Spécifiez tous les types de AWS ressources pris en charge.

*AWS::service_name::**

Spécifiez toutes les ressources prises en charge pour un service AWS spécifique.

AWS::service_name::resource_type

Spécifiez un type de AWS ressource spécifique, tel que *AWS::EC2::Instance* (toutes les instances EC2).

cloudformation:ResourceTypes

Types de ressource du modèle, comme *AWS::EC2::Instance*, que vous voulez associer à une politique. Utilisez cette condition pour contrôler les types de ressource dont les utilisateurs peuvent se servir lorsqu'ils créent ou mettent à jour d'une pile. Cette condition est comparée avec les types de ressource que les utilisateurs déclarent dans le paramètre *ResourceTypes*, qui est actuellement pris en charge uniquement pour les requêtes d' AWS CLI et d'API. Avec ce paramètre, les utilisateurs doivent spécifier tous les types de ressource qui se trouvent dans leur modèle. Pour plus d'informations sur le paramètre *ResourceTypes*, consultez l'action [CreateStack](#) dans la Référence API AWS CloudFormation .

Pour une liste des types de ressources, consultez le [Guide de référence des modèles CloudFormation](#).

Utilisez la convention de dénomination des ressources en trois parties pour spécifier les types de ressources que les utilisateurs peuvent utiliser, depuis toutes les ressources d'une organisation jusqu'à un type de ressource individuel.

*organization::**

Spécifiez tous les types de ressources pour une organisation donnée.

*organization::service_name::**

Spécifiez tous les types de ressources pour le service spécifié au sein d'une organisation donnée.

organization::service_name::resource_type

Spécifiez un type de ressource spécifique.

Par exemple :

AWS::*

Spécifiez tous les types de AWS ressources pris en charge.

AWS::*service_name*::*

Spécifiez toutes les ressources prises en charge pour un service AWS spécifique.

AWS::*service_name*::*resource_type*

Spécifiez un type de AWS ressource spécifique, tel que AWS::EC2::Instance (toutes les instances EC2).

Alexa::ASK::*

Spécifiez tous les types de ressources dans le kit de compétences Alexa.

Alexa::ASK::Skill

Spécifiez individuellement le type de ressource [Alexa::ASK::Skill](#).

Custom::*

Spécifiez toutes les ressources personnalisées.

Pour de plus amples informations, veuillez consulter [Créer une logique de provisionnement personnalisée avec des ressources personnalisées](#).

Custom::*resource_type*

Spécifiez un type de ressource personnalisé spécifique.

Pour de plus amples informations, veuillez consulter [Créer une logique de provisionnement personnalisée avec des ressources personnalisées](#).

`cloudformation:RoleARN`

Amazon Resource Name (ARN) du rôle de service IAM que vous voulez associer à une politique. Utilisez cette condition pour contrôler le rôle de service que les utilisateurs peuvent utiliser lorsqu'ils utilisent des piles ou des jeux de modifications.

`cloudformation:StackPolicyUrl`

URL de la politique de pile Amazon S3 que vous souhaitez associer à une politique. Utilisez cette condition pour contrôler les stratégies de pile que les utilisateurs peuvent associer à une pile pendant la création ou la mise à jour de cette dernière. Pour plus d'informations sur les politiques de pile, consultez [Empêcher les mises à jour des ressources de la pile](#).

Note

Pour vous assurer que les utilisateurs ne peuvent créer ou mettre à jour des piles qu'avec les politiques de pile que vous avez chargées, définissez le compartiment S3 en lecture seule pour ces utilisateurs.

`cloudformation:TemplateUrl`

URL d'un modèle Amazon S3 que vous voulez associer à une politique. Utilisez cette condition pour contrôler les modèles auxquels les utilisateurs peuvent se reporter lorsqu'ils créent ou mettent à jour des piles.

Note

Pour vous assurer que les utilisateurs ne peuvent créer ou mettre à jour des piles qu'avec les modèles que vous avez chargés, définissez également le compartiment S3 en lecture seule pour ces utilisateurs.

Note

Les conditions CloudFormation spécifiques suivantes s'appliquent aux paramètres d'API du même nom :

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`

- `cloudformation:StackPolicyUrl`
- `cloudformation:TemplateUrl`

Par exemple, s'applique `cloudformation:TemplateUrl` uniquement au `TemplateUrl` paramètre pour `CreateStackUpdateStack`, et `CreateChangeSet` APIs.

Pour des exemples de politiques IAM utilisant des clés de condition afin de contrôler l'accès, consultez [Exemples de politiques basées sur l'identité IAM pour CloudFormation](#).

Reconnaissance des ressources IAM dans les modèles CloudFormation

Avant de créer une pile, CloudFormation validez votre modèle. Lors de la validation, CloudFormation vérifie dans votre modèle les ressources IAM qu'il est susceptible de créer. Les ressources IAM, telles qu'un utilisateur disposant d'un accès complet, peuvent consulter et modifier n'importe quelle ressource de votre Compte AWS. Par conséquent, nous vous suggérons de passer en revue les autorisations associées à chaque ressource IAM avant de poursuivre. Vous éviterez ainsi de créer par erreur des ressources disposant d'autorisations que vous ne souhaitez pas accorder. Pour vous assurer que vous l'avez fait, vous devez reconnaître que le modèle contient ces ressources, en fournissant CloudFormation les fonctionnalités spécifiées avant de créer la pile.

Vous pouvez reconnaître les fonctionnalités des CloudFormation modèles à l'aide de la CloudFormation console, AWS Command Line Interface (AWS CLI) ou de l'API :

- Dans la CloudFormation console, sur la page Configurer les options de pile des assistants Create Stack ou Update Stack, sélectionnez Je reconnais que ce modèle peut créer des ressources IAM.
- Dans le AWS CLI, lorsque vous utilisez les [update-stack](#) commandes [create-stack](#), spécifiez la `CAPABILITY_NAMED_IAM` valeur `CAPABILITY_IAM` ou pour l'--capabilitiesoption. Si votre modèle comprend des ressources IAM, vous pouvez spécifier l'une ou l'autre de ces fonctionnalités. S'il comprend des noms personnalisés pour les ressources IAM, vous devez spécifier `CAPABILITY_NAMED_IAM`.
- Dans l'API, lorsque vous utilisez les actions [CreateStack](#) et [UpdateStack](#), spécifiez `Capabilities.member.1=CAPABILITY_IAM` ou `Capabilities.member.1=CAPABILITY_NAMED_IAM`. Si votre modèle comprend des ressources IAM, vous pouvez spécifier l'une ou l'autre de ces fonctionnalités. S'il comprend des noms personnalisés pour les ressources IAM, vous devez spécifier `CAPABILITY_NAMED_IAM`.

 Important

Si votre modèle contient des ressources IAM dont le nom est personnalisé, ne créez pas plusieurs piles avec le même modèle. De manière générale, les ressources IAM doivent être uniques au sein de votre compte. Si vous utilisez le même modèle pour créer plusieurs piles dans différentes régions, vos piles risquent de partager les mêmes ressources IAM au lieu d'avoir chacune une ressource unique. Les ressources partagées entre les piles peuvent avoir des conséquences imprévues et irréversibles. Par exemple, si vous supprimez ou mettez à jour des ressources IAM partagées dans une pile, vous modifierez les ressources des autres piles sans le vouloir.

Gestion des informations d'identification pour les applications exécutées dans des instances Amazon EC2

Si une application s'exécute sur une instance Amazon EC2 et doit effectuer des demandes pour obtenir des ressources AWS comme des compartiments Amazon S3 ou une table DynamoDB, l'application doit disposer d'informations d'identification de sécurité AWS . Toutefois, la distribution et intégration d'informations d'identification de sécurité à long terme dans toutes les instances que vous lancez peuvent s'avérer compliquées et présentent un risque de sécurité potentiel. Au lieu d'avoir recours à des informations d'identification à long terme, telles que les informations d'identification des utilisateurs IAM, nous vous recommandons de créer un rôle IAM associé à une instance Amazon EC2 lorsque l'instance sera lancée. L'application pourra ensuite obtenir des informations d'identification de sécurité temporaires à partir de cette instance Amazon EC2. Vous n'avez pas à intégrer d'informations d'identification à long terme dans l'instance. En outre, pour faciliter la gestion des informations d'identification, vous pouvez spécifier un rôle unique pour plusieurs instances Amazon EC2. Autrement dit; vous n'avez pas besoin de créer des identifiants uniques pour chacune d'elles.

Pour voir un extrait de modèle qui montre comment lancer une instance avec un rôle, consultez la page [Exemples de modèle pour les rôles IAM](#).

 Note

Les applications associées à des instances qui utilisent des informations d'identification de sécurité temporaires peuvent appeler n'importe quelle action CloudFormation . Toutefois, étant donné qu'il CloudFormation interagit avec de nombreux autres AWS services, vous

devez vérifier que tous les services que vous souhaitez utiliser prennent en charge les informations d'identification de sécurité temporaires. Pour une liste des services prenant en charge les informations d'identification de sécurité temporaires, consultez [Services AWS qui fonctionnent avec IAM](#) dans le Guide de l'utilisateur IAM.

Autorisation d'un accès temporaire (accès fédéré)

Dans certains cas, vous souhaitez peut-être accorder aux utilisateurs sans AWS informations d'identification un accès temporaire à votre Compte AWS. Plutôt que de créer et de supprimer des informations d'identification à long terme chaque fois que vous souhaitez accorder un accès temporaire, utilisez AWS Security Token Service (AWS STS). Par exemple, vous pouvez utiliser des rôles IAM. À partir d'un rôle IAM, vous pouvez créer, puis distribuer de nombreuses informations d'identification de sécurité temporaires par programmation (notamment une clé d'accès, une clé d'accès secrète et un jeton de sécurité). Ces informations d'identification ont une durée de vie limitée, elles ne peuvent donc pas être utilisées pour accéder à vos informations d'identification Compte AWS après leur expiration. Vous pouvez également créer plusieurs rôles IAM afin d'accorder à différents utilisateurs des niveaux d'autorisations distincts. Les rôles IAM sont utiles pour des scénarios comme les identités fédérées et l'authentification unique.

Une identité fédérée est une identité distincte que vous pouvez utiliser sur plusieurs systèmes. Les utilisateurs en entreprise disposant d'un système d'identité sur site (par exemple, LDAP ou Active Directory) peuvent s'en servir pour gérer toutes les authentifications. Une fois qu'un utilisateur a été authentifié, vous fournissez des informations d'identification de sécurité temporaires à partir de l'utilisateur ou du rôle IAM approprié. Par exemple, vous pouvez créer un rôle d'administrateur et un rôle de développeur, dans lesquels les administrateurs ont un accès complet au AWS compte et les développeurs sont autorisés à travailler uniquement avec des CloudFormation piles. Après authentification, un administrateur est autorisé à obtenir des informations d'identification de sécurité temporaires à partir du rôle administrateurs. Les développeurs, quant à eux, peuvent obtenir des informations d'identification de sécurité temporaires uniquement depuis le rôle développeurs.

Vous pouvez également accorder aux utilisateurs fédérés l'accès au AWS Management Console. Une fois que les utilisateurs s'authentifient via votre système d'identité sur site, vous pouvez créer une URL temporaire par programmation afin de leur permettre d'accéder directement à AWS Management Console. Lorsque les utilisateurs utilisent l'URL temporaire, ils n'ont pas besoin de se connecter AWS car ils ont déjà été authentifiés (authentification unique). En outre, comme l'URL repose sur les informations d'identification de sécurité temporaires des utilisateurs, les autorisations associées déterminent les actions qu'ils peuvent effectuer dans AWS Management Console.

Vous pouvez en utiliser plusieurs AWS STS APIs pour générer des informations d'identification de sécurité temporaires. Pour plus d'informations sur l'API à utiliser, consultez [Comparaison des informations d'identification AWS STS](#) dans le Guide de l'utilisateur IAM.

 Important

Vous ne pouvez pas utiliser IAM avec les informations d'identification de sécurité temporaires qui ont été générées à partir de l'API `GetFederationToken`. Si vous avez besoin de travailler avec IAM, utilisez plutôt les informations d'identification de sécurité temporaires d'un rôle.

CloudFormation interagit avec de nombreux autres AWS services. Lorsque vous utilisez des informations d'identification de sécurité temporaires avec CloudFormation, vérifiez que tous les services que vous souhaitez utiliser prennent en charge les informations d'identification de sécurité temporaires. Pour une liste des services prenant en charge les informations d'identification de sécurité temporaires, consultez [Services AWS qui fonctionnent avec IAM](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations, consultez les ressources associées dans le Guide de l'utilisateur IAM :

- [Scénarios courants pour les informations d'identification temporaires](#)
- [Permettre à un courtier d'identité personnalisé d'accéder à la AWS console](#)

Exemples de politiques basées sur l'identité IAM pour CloudFormation

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources CloudFormation. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles. Pour de plus amples informations, veuillez consulter [Définition de politiques basées sur l'identité IAM pour CloudFormation](#).

Les exemples suivants présentent des déclarations de politique que vous pouvez utiliser pour autoriser ou refuser l'autorisation d'utiliser une ou plusieurs CloudFormation actions.

Rubriques

- [Exiger une URL de modèle spécifique](#)
- [Refuser toutes les opérations CloudFormation d'importation](#)
- [Autoriser les opérations d'importation pour des types de ressources spécifiques](#)
- [Refuser les ressources IAM dans les modèles de pile](#)
- [Autoriser la création de piles avec des types de ressources spécifiques](#)
- [Contrôler l'accès en fonction des actions API modifiant les ressources](#)
- [Restriction des opérations sur les ensembles de piles en fonction de la région et des types de ressources](#)
- [Autoriser toutes les opérations du générateur IaC](#)

Exiger une URL de modèle spécifique

La politique suivante accorde les autorisations d'utiliser uniquement l'URL de modèle *https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template* pour créer ou mettre à jour une pile.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudformation:TemplateUrl": [
            "https://s3.amazonaws.com/amzn-s3-demo-bucket/
test.template"
          ]
        }
      }
    }
  ]
}
```

```
}
```

Refuser toutes les opérations CloudFormation d'importation

La politique suivante accorde les autorisations nécessaires pour effectuer toutes les CloudFormation opérations, à l'exception des opérations d'importation.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllStackOperations",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    },
    {
      "Sid": "DenyImport",
      "Effect": "Deny",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    }
  ]
}
```

Autoriser les opérations d'importation pour des types de ressources spécifiques

La politique suivante accorde les autorisations nécessaires pour toutes les opérations de pile, en plus des opérations d'importation uniquement sur les ressources spécifiées (dans cet exemple, `AWS::S3::Bucket`).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImport",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:ImportResourceTypes": [
            "AWS::S3::Bucket"
          ]
        }
      }
    }
  ]
}
```

Refuser les ressources IAM dans les modèles de pile

La politique suivante accorde les autorisations nécessaires pour créer des piles, mais refuse les demandes si le modèle de la pile inclut une ressource du service IAM. Cette politique oblige également les utilisateurs à spécifier le paramètre `ResourceTypes`, qui est uniquement disponible pour les requêtes d' AWS CLI et d'API. Cette politique utilise des instructions de refus explicites de sorte qu'elle reste en vigueur si une autre politique accorde des autorisations supplémentaires (une instruction de refus explicite prévaut toujours sur une instruction d'autorisation explicite).

JSON

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

```

    },
    {
      "Effect" : "Deny",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "ForAnyValue:StringLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        }
      }
    },
  ],
  {
    "Effect": "Deny",
    "Action" : [ "cloudformation:CreateStack" ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "cloudformation:ResourceTypes": "true"
      }
    }
  }
]
}

```

Autoriser la création de piles avec des types de ressources spécifiques

La politique suivante est similaire à l'exemple précédent. La politique accorde les autorisations nécessaires pour créer une pile, sauf si le modèle de la pile inclut une ressource du service IAM. Les utilisateurs doivent également spécifier le `ResourceTypes` paramètre, qui n'est disponible que pour AWS CLI les demandes d'API. Cette politique est plus simple, mais n'utilise pas d'instructions de refus explicites. Elle pourrait donc être supplantée par d'autres politiques, qui accorderaient des autorisations supplémentaires.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],

```

```

    "Resource" : "*",
    "Condition" : {
      "ForAllValues:StringNotLikeIfExists" : {
        "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
      },
      "Null":{
        "cloudformation:ResourceTypes": "false"
      }
    }
  }
]
}

```

Contrôler l'accès en fonction des actions API modifiant les ressources

La politique suivante accorde les autorisations nécessaires pour filtrer l'accès en fonction du nom d'une action API modifiant les ressources. Ceci est utilisé pour contrôler les utilisateurs APIs IAM qui peuvent ajouter ou supprimer des balises sur une pile ou un ensemble de piles. L'opération utilisée pour ajouter ou supprimer des balises doit être ajoutée comme valeur pour la clé de condition. La politique suivante accorde les autorisations TagResource et UntagResource à l'opération de modification CreateStack.

JSON

```

{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  ]
}

```

```
}]
}
```

Restriction des opérations sur les ensembles de piles en fonction de la région et des types de ressources

La politique suivante accorde des autorisations pour les ensembles de piles gérés par le service. Un utilisateur disposant de cette politique peut uniquement effectuer des opérations sur des ensembles de piles avec des modèles contenant des types de ressources Amazon S3 (`AWS::S3::*`) ou le type de ressource `AWS::SES::ConfigurationSet`. S'il est connecté au compte de gestion de l'organisation avec un identifiant `123456789012`, l'utilisateur peut également effectuer des opérations uniquement sur des ensembles de piles qui ciblent l'unité d'organisation dotée d'un identifiant `ou-1fsfsrdsdfrewr`, et ne peut effectuer des opérations que sur le jeu de piles doté d'un identifiant `stack-set-id` qui cible l'identifiant Compte AWS avec `987654321012`.

Les opérations de stack set échouent si le modèle de stack set contient des types de ressources autres que ceux spécifiés dans la stratégie, ou si les cibles de déploiement sont des unités d'organisation ou des comptes IDs autres que ceux spécifiés dans la stratégie pour les comptes de gestion et les ensembles de piles correspondants.

Ces restrictions de politique s'appliquent uniquement lorsque les opérations d'ensembles de piles ciblent les `us-east-1`, `us-west-2`, ou `eu-west-2` Régions AWS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/*",
        "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
        "arn:aws:cloudformation:us-west-2:111122223333:type/resource/AWS-SES-ConfigurationSet",

```

```

        "arn:aws:cloudformation:*:111122223333:stackset-target/
*/ou-1fsfsrdsdfrewr",
        "arn:aws:cloudformation:*:111122223333:stackset-target/stack-set-
id/444455556666"
    ],
    "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
            "cloudformation:TargetRegion": [
                "us-east-1",
                "us-west-2",
                "eu-west-1"
            ]
        }
    }
}

```

Autoriser toutes les opérations du générateur IaC

La politique suivante autorise l'accès aux CloudFormation actions liées à l'analyse des ressources du générateur iAC et à la gestion des modèles. La première instruction accorde les autorisations nécessaires pour décrire, répertorier et lancer des analyses de ressources. Elle permet également d'accéder à des autorisations supplémentaires requises (`cloudformation:GetResource`, `cloudformation:ListResources` et `cloudformation:ListTypes`) qui permettent au générateur IaC de récupérer des informations sur les ressources et les types de ressources disponibles. La deuxième instruction accorde toutes les autorisations nécessaires pour créer, supprimer, décrire, répertorier et mettre à jour les modèles générés.

Vous devez également accorder des autorisations de lecture pour les AWS services cibles à toute personne qui analysera les ressources avec le générateur IaC. Pour de plus amples informations, veuillez consulter [Autorisations IAM requises pour l'analyse des ressources](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ResourceScanningOperations",

```

```

        "Effect": "Allow",
        "Action": [
            "cloudformation:DescribeResourceScan",
            "cloudformation:GetResource",
            "cloudformation:ListResources",
            "cloudformation:ListResourceScanRelatedResources",
            "cloudformation:ListResourceScanResources",
            "cloudformation:ListResourceScans",
            "cloudformation:ListTypes",
            "cloudformation:StartResourceScan"
        ],
        "Resource": "*"
    },
    {
        "Sid": "TemplateGeneration",
        "Effect": "Allow",
        "Action": [
            "cloudformation:CreateGeneratedTemplate",
            "cloudformation>DeleteGeneratedTemplate",
            "cloudformation:DescribeGeneratedTemplate",
            "cloudformation:GetResource",
            "cloudformation:GetGeneratedTemplate",
            "cloudformation:ListGeneratedTemplates",
            "cloudformation:UpdateGeneratedTemplate"
        ],
        "Resource": "*"
    }
]
}

```

AWS politiques gérées pour AWS CloudFormation

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des [politiques gérées par le client](#) qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'un nouveau AWS service est lancé ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

AWS politique gérée : AWSCloudFormationFullAccess

Vous pouvez associer `AWSCloudFormationFullAccess` à vos utilisateurs, groupes et rôles.

Cette politique accorde des autorisations qui permettent un accès complet aux CloudFormation actions et aux ressources.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `cloudformation`— Permet aux principaux d'effectuer toutes les CloudFormation actions sur toutes les ressources.

Pour consulter les autorisations associées à cette politique, consultez [AWSCloudFormationFullAccess](#) le Guide de référence des politiques AWS gérées.

AWS politique gérée : AWSCloudFormationReadOnlyAccess

Vous pouvez associer `AWSCloudFormationReadOnlyAccess` à vos utilisateurs, groupes et rôles.

Cette politique accorde des autorisations permettant un accès en lecture seule aux CloudFormation ressources et aux actions.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- `cloudformation`— Permet aux principaux d'effectuer des CloudFormation actions en lecture seule, telles que la description des piles, la liste des ressources et l'affichage des modèles, mais n'autorise pas la création, la mise à jour ou la suppression de piles.

Pour consulter les autorisations associées à cette politique, consultez [AWS CloudFormationReadOnlyAccess](#) le Guide de référence des politiques AWS gérées.

CloudFormation mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CloudFormation 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 flux RSS sur la page Historique du CloudFormation document.

Modifier	Description	Date
AWS CloudFormationReadOnlyAccess – Mise à jour d'une stratégie existante	CloudFormation a ajouté de nouvelles autorisations pour autoriser <code>cloudformation:BatchDescribe*</code> les actions pour les opérations de description par lots.	30 janvier 2026
AWS CloudFormationReadOnlyAccess : mise à jour d'une politique existante	CloudFormation a ajouté de nouvelles autorisations pour autoriser les <code>cloudformation:Detect*</code> actions relatives aux capacités de détection de la dérive des piles.	13 novembre 2019
AWS CloudFormationReadOnlyAccess : mise à jour d'une politique existante	CloudFormation a ajouté de nouvelles autorisations pour autoriser <code>cloudformation:EstimateTemplateCost</code> , <code>cloudformation:Get*</code> , et <code>cloudformation:ValidateTemplate</code> des actions.	2 novembre 2017

Modifier	Description	Date
AWSCloudFormationFullAccess : nouvelle politique	CloudFormation a ajouté une nouvelle politique AWS gérée qui fournit un accès complet aux CloudFormation actions et aux ressources.	26 juillet 2019
AWSCloudFormationReadOnlyAccess : mise à jour d'une politique existante	CloudFormation ajout de nouvelles autorisations <code>cloudformation:DetectStackDrift</code> et <code>cloudformation:DetectStackResourceDrift</code> actions pour la détection de la dérive des piles.	6 février 2019
AWSCloudFormationReadOnlyAccess : nouvelle politique	CloudFormation a ajouté une nouvelle politique AWS gérée qui fournit un accès en lecture seule aux CloudFormation actions et aux ressources.	6 février 2015
CloudFormation a commencé à suivre les modifications	CloudFormation a commencé à suivre les modifications apportées AWS à ses politiques gérées.	6 février 2015

CloudFormation rôle de service

Un rôle de service est un rôle Gestion des identités et des accès AWS (IAM) qui permet CloudFormation d'appeler les ressources d'une pile en votre nom. Vous pouvez spécifier un rôle IAM qui permet à CloudFormation de créer, de mettre à jour ou de supprimer les ressources d'une pile. Par défaut, CloudFormation utilise une session temporaire générée à partir de vos informations d'identification utilisateur pour les opérations de stack. Si vous spécifiez un rôle de service, utilisez CloudFormation les informations d'identification de ce rôle.

Utilisez un rôle de service pour spécifier explicitement les actions que CloudFormation peut effectuer, qui ne sont pas toujours les mêmes que celles que vous ou les autres utilisateurs pouvez effectuer. Par exemple, vous pouvez avoir des privilèges administratifs, mais vous pouvez limiter CloudFormation l'accès aux seules EC2 actions Amazon.

Le rôle de service et sa politique d'autorisation sont créés à partir du service IAM. Pour plus d'informations sur la création d'un rôle de service, voir [Créer un rôle pour déléguer des autorisations à un AWS service](#) dans le Guide de l'utilisateur IAM. Spécifiez CloudFormation (`cloudformation.amazonaws.com`) en tant que service habilité à assumer le rôle.

Pour associer un rôle de service à une pile, spécifiez le rôle au moment où vous créez la pile. Pour en savoir plus, consultez [Configuration des options de pile](#). Vous pouvez également modifier le rôle de service lorsque vous mettez à jour la pile dans la console ou [DeleteStack](#) la pile via l'API. Avant de spécifier un rôle de service, vérifiez que vous êtes autorisé à le transmettre (`iam:PassRole`). L'autorisation `iam:PassRole` indique les rôles que vous pouvez utiliser. Pour plus d'informations, consultez la section [Accorder à un utilisateur l'autorisation de transmettre un rôle à un AWS service](#) dans le Guide de l'utilisateur IAM.

Important

Lorsque vous spécifiez un rôle de service, utilisez CloudFormation toujours ce rôle pour toutes les opérations effectuées sur cette pile. Un rôle de service attaché à une pile ne peut pas être supprimé après la création de la pile. Les autres utilisateurs disposant des autorisations nécessaires pour effectuer des opérations sur cette pile peuvent utiliser ce rôle, qu'ils disposent ou non de l'autorisation `iam:PassRole`. 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. Pour en savoir plus, consultez [Appliquer les autorisations de moindre privilège](#) dans le Guide de l'utilisateur IAM.

Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser

ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui CloudFormation accordent un autre service à une ressource spécifique, telle qu'une CloudFormation extension. 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.

Assurez-vous que la valeur de `aws:SourceArn` est un ARN de la ressource qui CloudFormation stocke.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:cloudformation:*:123456789012:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID de compte, vous devez utiliser les deux clés de contexte globales pour limiter les autorisations.

L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale CloudFormation pour éviter le problème de confusion des adjoints.

Exemple de politique d'approbation utilisant les clés de condition **aws:SourceArn** et **aws:SourceAccount**

Pour les services de CloudFormation registre, appelez AWS Security Token Service (AWS STS) pour assumer un rôle de service dans votre compte. Ce rôle est configuré pour `ExecutionRoleArn` dans l'opération [RegisterType](#) et `LogRoleArn` est défini dans l'opération [LoggingConfig](#). Pour de plus amples informations, veuillez consulter [Configurer un rôle d'exécution avec des autorisations IAM et une politique de confiance pour l'accès public aux extensions](#).

Cet exemple de politique de confiance dans les rôles utilise des instructions de condition pour limiter la `AssumeRole` capacité du rôle de service aux seules actions portant sur l'

CloudFormation extension spécifiée dans le compte spécifié. Les conditions `aws:SourceArn` et `aws:SourceAccount` sont évaluées indépendamment. Toute demande d'utilisation de la fonction du service doit répondre aux deux conditions.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/Organization-Service-Resource"
        }
      }
    }
  ]
}
```

Informations supplémentaires

Par exemple, les politiques qui utilisent les clés de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et les clés de contexte pour un rôle de service utilisé par StackSets, voir [Configurez des clés globales pour atténuer les problèmes de député confus](#).

Pour plus d'informations, consultez [Modification d'une politique d'approbation de rôle](#) dans le Guide de l'utilisateur IAM.

Transfert des demandes de sessions d'accès (FAS) et évaluation des autorisations

Lors de la création, de la mise à jour et de la suppression de CloudFormation piles, les utilisateurs peuvent éventuellement spécifier un ARN de rôle IAM. Si aucun rôle n'est fourni, CloudFormation utilise son mécanisme de service par défaut pour interagir avec d'autres AWS services. Dans ce scénario, l'appelant doit disposer des autorisations nécessaires pour les ressources gérées. Par ailleurs, lorsqu'un utilisateur fournit son propre rôle IAM, il CloudFormation assume ce rôle pour effectuer des interactions de service en son nom.

Que l'utilisateur fournisse ou non un rôle IAM, il CloudFormation génère un nouveau jeton FAS délimité pour chaque opération sur les ressources. Par conséquent, les [clés de condition liées au FAS](#), y compris `aws:ViaAWSService`, sont renseignées dans les deux scénarios.

L'utilisation du FAS influe sur la manière dont les politiques IAM sont évaluées pendant CloudFormation les opérations. Lors de la création d'une pile à l'aide d'un modèle qui inclut des ressources affectées par des clés de condition liées au FAS, des refus d'autorisation peuvent se produire.

Exemple de politique IAM

Tenez compte de la politique IAM suivante. `Statement2` empêchera systématiquement la création d'une `AWS::KMS::Key` ressource dans CloudFormation. La restriction sera appliquée de manière cohérente, qu'un rôle IAM soit fourni ou non pendant l'opération de pile. Cela est dû au fait que la clé de condition `aws:ViaAWSService` est toujours définie sur `true` en raison de l'utilisation du FAS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```

    },
    {
      "Sid": "Statement2",
      "Effect": "Deny",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    }
  ]
}

```

Exemple de modèle de pile

Par exemple, lorsqu'un utilisateur crée une pile à l'aide du modèle suivant, `aws:ViaAWSService` est défini sur `true`, et les autorisations de rôle seront remplacées par la politique FAS. La création de la pile sera affectée par `Statement2` de la politique IAM, qui refuse l'action `CreateKey`. Cela entraîne une erreur de refus d'autorisation.

```

Resources:
  myPrimaryKey:
    Type: AWS::KMS::Key
    Properties:
      Description: An example multi-Region primary key
      KeyPolicy:
        Version: '2012-10-17'
        Id: key-default-1
        Statement:
          - Sid: Enable IAM User Permissions
            Effect: Allow
            Principal:
              AWS: !Join
                - '-'
                - 'arn:aws:iam::'
                - !Ref AWS::AccountId

```

```
- ':root'  
Action: kms:*  
Resource: '*'
```

Pour plus d'informations sur le FAS, consultez la section [Transférer les sessions d'accès](#) dans le Guide de l'utilisateur IAM.

Note

La plupart des ressources adoptent ce comportement. Toutefois, si vous rencontrez un succès ou un échec inattendu lors de la création, de la mise à jour ou de la suppression d'une ressource, et que votre politique IAM inclut des clés de condition liées à FAS, il est probable que la ressource en question appartienne à un petit sous-ensemble de ressources qui ne suivent pas ce modèle standard.

Journalisation des appels d' CloudFormation API avec AWS CloudTrail

CloudFormation est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans CloudFormation. CloudTrail capture tous les appels d'API CloudFormation sous forme d'événements, y compris les appels depuis la CloudFormation console et les appels de code vers le CloudFormation APIs. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour CloudFormation. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite CloudFormation, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Rubriques

- [CloudFormation informations dans CloudTrail](#)
- [Comprendre les entrées du fichier CloudFormation journal](#)

CloudFormation informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans CloudFormation, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. 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](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour CloudFormation, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions . Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour en savoir plus, consultez :

- [Présentation de la création d'un journal d'activité](#)
- [CloudTrail services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Toutes les CloudFormation actions sont enregistrées CloudTrail et documentées dans la [référence de l'AWS CloudFormation API](#). Par exemple, les appels aux ListStacks sections CreateStackDeleteStack, et génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#) .

Comprendre les entrées du fichier CloudFormation journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'CreateStack opération. Cette opération a été effectuée par un utilisateur IAM prénommé Alice.

Note

Seuls les noms de clé des paramètre d'entrée sont consignés. Les valeurs des paramètres ne sont pas consignées.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHJKLMNOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:02:43Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "CreateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "tags": [
      {
        "key": "test",
        "value": "tag"
      }
    ]
  }
}
```

```

    }
  ],
  "stackName": "my-test-stack",
  "disableRollback": true,
  "parameters": [
    {
      "parameterKey": "password"
    },
    {
      "parameterKey": "securitygroup"
    }
  ]
},
"responseElements": {
  "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/a38e6a60-b397-11e3-b0fc-08002755629e"
},
"requestID": "9f960720-b397-11e3-bb75-a5b75389b02d",
"eventID": "9bf6cfb8-83e1-4589-9a70-b971e727099b"
}

```

L'exemple suivant montre qu'Alice a appelé l'opération UpdateStack au niveau de la pile my-test-stack :

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:04:29Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "UpdateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "templateURL",
    "parameters": [

```

```

    {
      "parameterKey": "password"
    },
    {
      "parameterKey": "securitygroup"
    }
  ],
  "stackName": "my-test-stack"
},
"responseElements": {
  "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/
a38e6a60-b397-11e3-b0fc-08002755629e"
},
"requestID": "def0bf5a-b397-11e3-bb75-a5b75389b02d",
"eventID": "637707ce-e4a3-4af1-8edc-16e37e851b17"
}

```

L'exemple suivant montre qu'Alice a appelé l'opération `ListStacks`.

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmnopq",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "ListStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
  "eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}

```

L'exemple suivant montre qu'Alice a appelé l'opération `DescribeStacks` au niveau de la pile `my-test-stack`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
  "eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

L'exemple suivant montre qu'Alice a appelé l'opération `DeleteStack` au niveau de la pile `my-test-stack`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DeleteStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",

```

```
"requestParameters": {  
  "stackName": "my-test-stack"  
},  
"responseElements": null,  
"requestID": "42dae739-b398-11e3-bb75-a5b75389b02d",  
"eventID": "4965eb38-5705-4942-bb7f-20ebe79aa9aa"  
}
```

Sécurité de l'infrastructure dans CloudFormation

En tant que service géré, AWS CloudFormation 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 CloudFormation 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.

Vous pouvez appeler ces opérations d'API depuis n'importe quel emplacement réseau, mais CloudFormation elles prennent en charge les politiques d'accès basées sur les ressources, qui peuvent inclure des restrictions basées sur l'adresse IP source. Vous pouvez également utiliser des CloudFormation politiques pour contrôler l'accès depuis des points de terminaison Amazon Virtual Private Cloud (Amazon VPC) spécifiques ou spécifiques. VPCs En fait, cela isole l'accès réseau à une CloudFormation ressource donnée uniquement du VPC spécifique au sein AWS du réseau.

Résilience dans CloudFormation

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont

davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Validation de conformité pour CloudFormation

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

Analyse de configuration et de vulnérabilité dans CloudFormation

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous, notre client. Pour plus d'informations, consultez le [modèle de responsabilité AWS partagée](#).

Bonnes pratiques de sécurité pour CloudFormation

CloudFormation fournit un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

Rubriques

- [Utilisation du contrôle d'accès IAM](#)
- [N'incorporez pas d'informations d'identification dans vos modèles](#)

- [AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels](#)

Utilisation du contrôle d'accès IAM

IAM est un AWS service que vous pouvez utiliser pour gérer les utilisateurs et leurs autorisations. AWS Vous pouvez utiliser IAM with CloudFormation pour spécifier les CloudFormation actions que les utilisateurs peuvent effectuer, telles que l'affichage des modèles de piles, la création de piles ou la suppression de piles. En outre, toute personne gérant des CloudFormation piles aura besoin d'autorisations pour accéder aux ressources contenues dans ces piles. Par exemple, si les utilisateurs souhaitent lancer CloudFormation , mettre à jour ou résilier des EC2 instances Amazon, ils doivent être autorisés à lancer les EC2 actions Amazon pertinentes.

Dans la plupart des cas, les utilisateurs ont besoin d'un accès complet pour gérer toutes les ressources dans un modèle. CloudFormation effectue des appels pour créer, modifier et supprimer ces ressources en leur nom. Pour séparer les autorisations entre un utilisateur et le CloudFormation service, utilisez un rôle de service. CloudFormation utilise la politique du rôle de service pour effectuer des appels au lieu de la politique de l'utilisateur. Pour de plus amples informations, veuillez consulter [CloudFormation rôle de service](#).

N'incorporez pas d'informations d'identification dans vos modèles

Plutôt que d'intégrer des informations sensibles dans vos CloudFormation modèles, nous vous recommandons d'utiliser des références dynamiques dans votre modèle de pile.

Les références dynamiques constituent un moyen compact et puissant de référencer des valeurs externes stockées et gérées dans d'autres services, tels que le magasin de paramètres de AWS Systems Manager ou le gestionnaire de AWS secrets. Lorsque vous utilisez une référence dynamique, CloudFormation récupère la valeur de la référence spécifiée lorsque cela est nécessaire lors des opérations de pile et de modification des ensembles de modifications, et transmet la valeur à la ressource appropriée. Cependant, CloudFormation ne stocke jamais la valeur de référence réelle. Pour de plus amples informations, veuillez consulter [Obtenir les valeurs stockées dans d'autres services à l'aide de références dynamiques](#).

[AWS Secrets Manager](#) vous aider à chiffrer, stocker et récupérer en toute sécurité des informations d'identification pour vos bases de données et d'autres services. [AWS Systems Manager Parameter Store](#) offre un stockage sécurisé et hiérarchique de la gestion des données de configuration.

Pour plus d'informations sur la définition des paramètres de modèle, consultez [CloudFormation Parameterssyntaxe du modèle](#).

AWS CloudTrail À utiliser pour enregistrer CloudFormation les appels

AWS CloudTrail suit toute personne effectuant des appels d' CloudFormation API dans votre Compte AWS. Les appels d'API sont enregistrés chaque fois que quelqu'un utilise l' CloudFormation API, la CloudFormation console, une console principale ou des CloudFormation AWS CLI commandes. Activez la journalisation et spécifiez un compartiment Amazon S3 pour y stocker les journaux. De cette façon, le cas échéant, vous pourrez déterminer qui a effectué quel appel CloudFormation dans votre compte. Pour de plus amples informations, veuillez consulter [Journalisation des appels d' CloudFormation API avec AWS CloudTrail](#).

Accès CloudFormation via un point de terminaison d'interface (AWS PrivateLink)

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et. CloudFormation Vous pouvez y accéder CloudFormation comme s'il se trouvait dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou Direct Connect de connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour y accéder. CloudFormation

Vous établissez cette connexion privée en créant un point de terminaison d'interface optimisé par AWS PrivateLink. Nous créons une interface réseau de point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par le demandeur qui servent de point d'entrée pour le trafic destiné à CloudFormation.

CloudFormation prend en charge les appels à toutes ses actions d'API via le point de terminaison de l'interface.

Considérations relatives aux points de CloudFormation terminaison VPC

Avant de configurer un point de terminaison d'interface, assurez-vous d'abord de remplir les conditions requises dans la rubrique [Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface](#) du Guide.AWS PrivateLink

Les conditions préalables et considérations supplémentaires suivantes s'appliquent lors de la configuration d'un point de terminaison d'interface pour CloudFormation :

- Si des ressources au sein de votre VPC doivent répondre à une demande de ressources personnalisée ou à une condition d'attente, assurez-vous qu'elles ont accès aux compartiments

Amazon CloudFormation S3 spécifiques requis. CloudFormation dispose de compartiments S3 dans chaque région pour surveiller les réponses à une demande de [ressource personnalisée](#) ou à une [condition d'attente](#). Si un modèle inclut des ressources personnalisées ou des conditions d'attente dans un VPC, la politique de point de terminaison du VPC doit permettre aux utilisateurs d'envoyer des réponses aux compartiments suivants :

- Pour les ressources personnalisées, autorisez le trafic vers le compartiment `cloudformation-custom-resource-response-region`. Lorsque vous utilisez des ressources personnalisées, Région AWS les noms ne contiennent pas de tirets. Par exemple, `uswest2`.
- Pour les conditions d'attente, autorisez le trafic vers le compartiment `cloudformation-waitcondition-region`. Lorsque des conditions d'attente sont utilisées, Région AWS les noms contiennent des tirets. Par exemple, `us-west-2`.

Si la politique du point de terminaison bloque le trafic vers ces compartiments, CloudFormation ne reçoit pas de réponses et que l'opération de stack échoue. Par exemple, si une ressource d'un VPC situé dans la région `us-west-2` doit répondre à une condition d'attente, celle-ci doit être en mesure d'envoyer une réponse au compartiment `cloudformation-waitcondition-us-west-2`.

Pour une liste des Régions AWS endroits CloudFormation actuellement disponibles, consultez la page des [CloudFormation points de terminaison et des quotas](#) dans le Référence générale d'Amazon Web Services.

- Les points de terminaison VPC ne prennent actuellement pas en charge les demandes interrégionales. Assurez-vous de créer votre point de terminaison dans la même région que celle à laquelle vous prévoyez d'envoyer vos appels d'API. CloudFormation
- Les points de terminaison d'un VPC prennent uniquement en charge le DNS fourni par Amazon via Route 53. Si vous souhaitez utiliser votre propre DNS, vous pouvez utiliser le transfert DNS conditionnel. Pour plus d'informations, consultez [Ensembles d'options DHCP dans Amazon VPC](#) du Guide de l'utilisateur Amazon VPC.
- Le groupe de sécurité attaché au point de terminaison d'un VPC doit autoriser les connexions entrantes sur le port 443 à partir du sous-réseau privé du VPC.

Création d'un point de terminaison VPC d'interface pour CloudFormation

Vous pouvez créer un point de terminaison VPC pour CloudFormation utiliser la console Amazon VPC ou le (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez [Créer un point de terminaison d'un VPC](#) dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour CloudFormation utiliser le nom de service suivant :

- `com.amazonaws.region.cloudformation` — Crée un point de terminaison pour les opérations d'CloudFormation API.

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API à CloudFormation l'aide de son nom DNS régional par défaut. Par exemple, `cloudformation.us-east-1.amazonaws.com`.

Régions AWS Lorsque les points de terminaison spécifiques à FIPS sont pris en charge, vous pouvez également créer un point de terminaison d'interface CloudFormation utilisant le nom de service suivant :

- `com.amazonaws.region.cloudformation-fips` — Crée un point de terminaison pour l'CloudFormation API conforme à la [norme fédérale de traitement de l'information](#) (FIPS) 140-2.

Pour une liste complète des points de CloudFormation terminaison, consultez la section [Points de CloudFormation terminaison et quotas](#) dans le. Référence générale d'Amazon Web Services

Création d'une politique de point de terminaison VPC pour CloudFormation

Une politique de point de terminaison est une ressource IAM que vous pouvez attacher à votre point de terminaison d'interface. La politique de point de terminaison par défaut autorise un accès complet CloudFormation via le point de terminaison de l'interface. Pour contrôler l'accès autorisé CloudFormation depuis votre VPC, associez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Les principaux qui peuvent effectuer des actions (Comptes AWS, utilisateurs IAM et rôles IAM).
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

Pour plus d'informations, consultez [Contrôle de l'accès aux points de terminaison d'un VPC à l'aide de politiques de point de terminaison](#) dans le AWS PrivateLink .

Exemple : politique de point de terminaison VPC pour les actions CloudFormation

Voici un exemple de politique de point de terminaison pour CloudFormation. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux CloudFormation actions répertoriées pour tous les principaux sur toutes les ressources. L'exemple suivant refuse à tous les utilisateurs l'autorisation de créer des piles via le point de terminaison VPC et autorise un accès complet à toutes les autres actions du service. CloudFormation

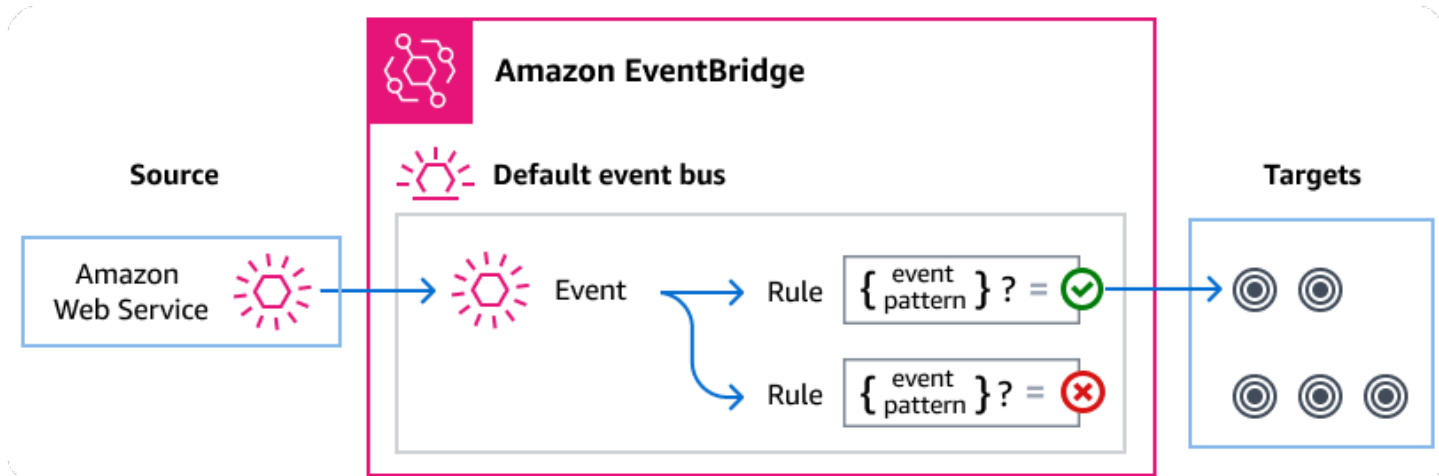
```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Surveillance CloudFormation et synchronisation des événements avec Git avec EventBridge

Amazon EventBridge est un service sans serveur qui utilise des événements pour connecter les composants de l'application entre eux, ce qui vous permet de créer plus facilement des applications évolutives pilotées par des événements. Une architecture pilotée par les événements est un style de création de systèmes logiciels faiblement couplés qui fonctionnent ensemble en émettant des événements et en y répondant. Les événements représentent une modification d'une ressource ou d'un environnement.

Comme c'est le cas pour de nombreux AWS services, CloudFormation génère et envoie des événements au bus d'événements EventBridge par défaut, qui est automatiquement provisionné dans chaque Compte AWS service. Un bus d'événements est un routeur qui reçoit des événements et les transmet à zéro ou plusieurs destinations, ou cibles. Les règles que vous spécifiez pour le bus d'événements évaluent les événements à mesure qu'ils arrivent. Chaque règle vérifie si un événement correspond au modèle d'événement de la règle. Si l'événement correspond, le bus d'événements envoie l'événement à la ou aux cibles spécifiées.

Pour plus d'informations, consultez [Getting started with Amazon EventBridge](#) dans le guide de EventBridge l'utilisateur Amazon.



Rubriques

- [CloudFormation et présentation des événements de synchronisation Git](#)
- [EventBridge Autorisations Amazon](#)
- [Création d'un modèle d'événement personnalisé pour une EventBridge règle](#)

- [CloudFormation référence détaillée des événements](#)

CloudFormation et présentation des événements de synchronisation Git

CloudFormation envoie des événements à EventBridge chaque fois qu'une opération de création, de mise à jour, de suppression ou de détection de dérive est effectuée sur une pile. CloudFormation envoie également des événements EventBridge pour modifier le statut des ensembles de piles et des instances d'ensembles de piles. Vous pouvez utiliser des EventBridge règles pour acheminer les événements vers les cibles que vous avez définies. La livraison de ces événements est garantie, et il se peut qu'ils ne soient pas transmis dans l'ordre.

Étant donné que les CloudFormation événements représentent des modifications apportées aux piles ou aux ensembles de piles et à leurs ressources, vous pouvez les utiliser pour lancer des flux de travail associés à des événements respectifs. Par exemple :

- Créez des balises spécifiques à la pile ou à l'ensemble de piles sur toutes les ressources provisionnées par CloudFormation.
- Établissez une association entre une CloudFormation pile ou un ensemble de piles et un Amazon WorkSpaces Application Manager (Amazon WAM).
- Spécifiez une association avec un AppRegistry pour la pile ou l'ensemble de piles créé.

Les événements suivants sont générés CloudFormation et envoyés au bus d'événements par défaut dans EventBridge. Pour de plus amples informations, veuillez consulter [???](#).

Type d'événement	Description	Source de l'événement
Changement de statut des ressources	Toute mise à jour effectuée sur une pile qui modifie les propriétés sous-jacentes des ressources. Pour obtenir la liste complète des types de AWS ressources pris en charge, consultez la référence des types de AWS ressources et de propriétés.	AWS CloudFormation

Type d'événement	Description	Source de l'événement
Changement d'état de la pile	Représente une modification d'une pile donnée. Pour plus de détails sur le code, voir Codes d'état de la pile .	AWS CloudFormation
Changement d'état de la détection d'écart	Représente une mise à jour de détection d'écart initiée par l'utilisateur sur une pile donnée. Pour obtenir la liste complète des types entièrement mutables et immuables qui prennent en charge la détection d'écart, consultez Prise en charge des types de ressources .	AWS CloudFormation
StackSetChangement de statut	Représente un changement d'état d'un ensemble de piles donné.	AWS CloudFormation
StackSet Modification du statut de l'instance Stack	Représente un changement de statut d'une instance de StackSet pile spécifique. Pour plus de détails sur le code, voir Codes d'état de l'instance de pile .	AWS CloudFormation
StackSet état de l'opération	Représente le changement de statut d'une StackSet opération donnée. Pour plus de détails sur le code, voir StackSets codes de statut .	AWS CloudFormation

En outre, AWS CloudFormation Git sync envoie des événements pour les changements de statut pour les synchronisations de référentiels et les synchronisations de ressources vers. EventBridge

Les événements de synchronisation Git suivants sont générés CodeConnections et envoyés au bus d'événements par défaut dans EventBridge. Pour de plus amples informations, veuillez consulter [???](#).

Type d'événement	Description	Source de l'événement
Modification de l'état de synchronisation du référentiel	Représente une modification d'un référentiel Git.	AWS CodeConnections
Modification de l'état de synchronisation des ressources	Représente une modification d'une synchronisation des ressources Git.	AWS CodeConnections

EventBridge Autorisations Amazon

CloudFormation ne nécessite aucune autorisation supplémentaire pour diffuser des événements à EventBridge. Les événements contiennent des informations déjà disponibles via les opérations CloudFormation de l'API.

Les cibles que vous spécifiez peuvent nécessiter des autorisations ou une configuration spécifiques. Pour plus de détails sur l'utilisation de services spécifiques pour les cibles, consultez [EventBridge les cibles Amazon](#) dans le guide de EventBridge l'utilisateur Amazon.

Création d'un modèle d'événement personnalisé pour une EventBridge règle

Vous pouvez trouver plusieurs modèles prédéfinis dans les événements de synchronisation EventBridge for CloudFormation et Git. Cela simplifie la création d'un modèle d'événement. Au lieu d'écrire vos propres modèles d'événements, vous pouvez sélectionner des valeurs de champ dans un formulaire et EventBridge générer le modèle pour vous. Vous pouvez créer une nouvelle règle à l'aide de l'un de ces modèles d'événements prédéfinis ou créer votre propre modèle d'événement personnalisé.

Lorsqu'un service CloudFormation fournit un événement au bus d'événements par défaut, EventBridge utilise le modèle d'événement défini dans votre règle pour déterminer si l'événement doit être transmis aux cibles de la règle. Un modèle d'événement correspond aux données des CloudFormation événements souhaités.

Chaque modèle d'événement est un objet JSON qui contient :

- Attribut `source` qui identifie le service d'où provient l'événement. Par exemple, `aws.cloudformation` ou `aws.codeconnections`.
- (Facultatif) : un attribut `detail-type` qui contient un tableau des types d'événements à associer.
- (Facultatif) : un attribut `detail` qui contient toute autre donnée d'événement à rechercher.

Par exemple, l'ID de la pile, les ressources utilisées, l'état des différentes ressources, et d'autres données relatives à un type d'événement particulier.

Par exemple, le modèle d'événement suivant correspond à tous les événements de changement d'état des ressources :

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

Alors que le modèle d'événement suivant utilise les données détaillées des événements pour correspondre uniquement aux événements de changement d'état des ressources qui CloudFormation créent une nouvelle `AWS::SNS::Topic` ressource `AWS::S3::Bucket` ou une nouvelle ressource :

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

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

CloudFormation référence détaillée des événements

Tous les événements des AWS services ont un ensemble commun de champs contenant des métadonnées relatives à l'événement, telles que le AWS service à l'origine de l'événement, l'heure à laquelle l'événement a été généré, le compte et la région dans lesquels l'événement a eu lieu, etc.

Pour les définitions de ces champs généraux, voir les [métadonnées des événements de service AWS](#) dans la référence des événements AWS .

En outre, chaque événement possède un champ `detail` qui contient des données spécifiques à cet événement en particulier. La référence ci-dessous définit les champs de détail des différents CloudFormation événements.

Lorsque vous l'utilisez EventBridge pour sélectionner et gérer CloudFormation des événements, il est utile de garder à l'esprit les points suivants :

- Ce champ `source` spécifie la source de l'événement.

Par exemple, `aws.cloudformation` ou `aws.codeconnections`.

- Le champ `detail-type` indique le type d'événement.

Par exemple, `CloudFormation Resource Status Change` ou `CloudFormation Drift Detection Status Change`.

- Le champ `detail` contient les données spécifiques à cet événement en particulier.

Par exemple, l'ID de la pile, les ressources utilisées, l'état des différentes ressources, et d'autres données relatives à un type d'événement particulier.

Pour plus d'informations sur la création de modèles d'événements permettant aux règles de correspondre aux CloudFormation événements, consultez la section [Modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations sur les événements et leur EventBridge traitement, consultez le guide de l'EventBridge utilisateur Amazon consacré aux [EventBridge événements](#).

Rubriques

- [Détails de l'événement « Modification de l'état des ressources »](#)
- [Détails de l'événement « Changement d'état de la pile »](#)
- [État de détection d'écart Modifier les détails de l'événement](#)
- [StackSet Détails de l'événement de changement de statut](#)
- [StackSet Stack Instance Status Modifier le détail de l'événement](#)
- [StackSet État de l'opération Modifier le détail de l'événement](#)
- [État de synchronisation du référentiel Modifier le détail de l'événement](#)

- [Détails de l'événement « Modification de l'état de synchronisation des ressources »](#)

Détails de l'événement « Modification de l'état des ressources »

Vous trouverez ci-dessous les champs détaillés pour les événements de modification de l'état des ressources.

Les `detail-type` champs `source` et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```
{
  . . . ,
  "detail-type": "CloudFormation Resource Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-id" : "string",
    "logical-resource-id" : "string",
    "physical-resource-id": "string",
    "status-details": {
      "status": "string",
      "status-reason": "string"
    },
    "resource-type": "string",
    "client-request-token": "string"
  }
}
```

`detail-type`

Identifie le type d'événement.

Pour les événements d'état des ressources, cette valeur est `CloudFormation Resource Status Change`.

`source`

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements d'état des ressources, ces données incluent :

`stack-id`

L'ID unique de la pile auquel la pile est associée.

`logical-resource-id`

Le nom logique de la ressource, tel que spécifié dans le modèle.

`physical-resource-id`

Le nom ou l'identifiant unique qui correspond à l'ID d'instance physique d'une ressource prise en charge par CloudFormation.

`status-details`

`status`

État de la ressource.

`status-reason`

Raison de l'état de la ressource.

`resource-type`

Type de ressource. Par exemple, `AWS::S3::Bucket`.

`client-request-token`

Un jeton d'accès utilisé pour appeler l'API. Tous les événements lancés par une opération de pile donnée se voient attribuer le même jeton de demande client, que vous pouvez utiliser pour suivre les opérations. Les opérations de pile lancées depuis la console utilisent le format de jeton Console- StackOperation -ID, qui vous permet d'identifier facilement l'opération de pile. Par exemple, si vous créez une pile à l'aide de la console, chaque événement de pile résultant reçoit le même jeton au format suivant : `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Exemple Exemple : événement de changement d'état des ressources

Voici un exemple d'événement d'état des ressources. Cet événement indique que CloudFormation a réussi à créer la ressource demandée, un compartiment Amazon S3, dans la pile spécifiée.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Resource Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
  ],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "logical-resource-id": "my-s3-bucket",
    "physical-resource-id": "arn:aws:s3::my-s3-bucket-us-east-1",
    "status-details": {
      "status": "CREATE_COMPLETE",
      "status-reason": ""
    },
    "resource-type": "AWS::S3::Bucket",
    "client-request-token": ""
  }
}
```

Détails de l'événement « Changement d'état de la pile »

Vous trouverez ci-dessous les champs détaillés pour les événements de changement d'état de la pile.

Les `detail-type` champs `source` et `sont` inclus car ils contiennent des valeurs spécifiques pour les événements.

```
{
  . . . ,
  "detail-type": "CloudFormation Stack Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
```

```
"stack-id":"string",
"status-details":{
  "status":"string",
  "status-reason":"string"
},
"client-request-token":"string"
}
```

detail-type

Identifie le type d'événement.

Pour les événements d'état de la pile, cette valeur est `CloudFormation Stack Status Change`.

source

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements d'état de la pile, ces données incluent :

stack-id

L'ID unique de la pile à laquelle la pile est associée.

status-details

status

État de la pile.

Pour une liste complète des codes d'état de la pile, consultez [Codes d'état de la pile](#).

status-reason

Raison de l'état de la ressource.

client-request-token

Un jeton d'accès utilisé pour appeler l'API. Tous les événements lancés par une opération de pile donnée se voient attribuer le même jeton de demande client, que vous pouvez utiliser pour

suivre les opérations. Les opérations de pile lancées depuis la console utilisent le format de jeton Console- StackOperation -ID, qui vous permet d'identifier facilement l'opération de pile. Par exemple, si vous créez une pile à l'aide de la console, chaque événement de pile résultant reçoit le même jeton au format suivant : Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Exemple Exemple : événement d'état de pile

Voici un exemple d'événement d'état de pile, où la pile demandée CloudFormation a été créée avec succès `teststack`.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Stack Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
  ],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "status-details": {
      "status": "CREATE_COMPLETE",
      "status-reason": ""
    },
    "client-request-token": ""
  }
}
```

État de détection d'écart Modifier les détails de l'événement

Vous trouverez ci-dessous les champs de détails des événements de détection des écarts de pile.

Les `detail-type` champs `source` et `sont` inclus car ils contiennent des valeurs spécifiques pour les événements.

```
{
```

```
. . . ,
"detail-type":"CloudFormation Drift Detection Status Change",
"source":"aws.cloudformation",
. . . ,
"detail":{
  "stack-id":"string",
  "stack-drift-detection-id":"string",
  "status-details":{
    "stack-drift-status":"string",
    "detection-status":"string"
  },
  "drift-detection-details":{
    "drifted-stack-resource-count":integer
  },
  "client-request-token":"string"
}
```

detail-type

Identifie le type d'événement.

Pour les événements de détection de l'écart de pile, cette valeur est CloudFormation Drift Detection Status Change.

source

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements de détection de l'écart de pile, ces données incluent :

stack-id

L'ID unique de la pile auquel la pile est associée.

stack-drift-detection-id

L'ID de détection des écarts de pile.

status-details

stack-drift-status

État d'écart de la pile.

detection-status

État de l'opération de détection des écarts.

drift-detection-details

drifted-stack-resource-count

Texte prédéfini – Pour ce type de liste (), vous créez un fichier de type texte brut délimité par des lignes qui répertorie le texte spécifique à ignorer, et vous stockez le fichier dans un compartiment Amazon Simple Storage Service (Amazon S3). Lorsque la valeur est -1, la détection des écarts est en cours. Tous les autres entiers non négatifs représentent le nombre réel de ressources présentant un écart.

client-request-token

Un jeton d'accès utilisé pour appeler l'API. Tous les événements lancés par une opération de pile donnée se voient attribuer le même jeton de demande client, que vous pouvez utiliser pour suivre les opérations. Les opérations de pile lancées depuis la console utilisent le format de jeton Console- StackOperation -ID, qui vous permet d'identifier facilement l'opération de pile. Par exemple, si vous créez une pile à l'aide de la console, chaque événement de pile résultant reçoit le même jeton au format suivant : Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Exemple Exemple : événement de détection d'un écart de pile

Voici un exemple d'événement de détection d'un écart de pile. Cet événement indique que la détection de dérive sur la pile spécifiée CloudFormation a été terminée et que l'état de dérive de la pile est actuellement DRIFTED dû à une ressource dérivée.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Drift Detection Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
```



```

"resources": ["string"],
"detail":{
  "stack-id":"arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
  "stack-drift-detection-id":"624af370-311a-11e8-b6b7-500cexample",
  "status-details":{
    "stack-drift-status":"DRIFTED",
    "detection-status":"DETECTION_COMPLETE"
  },
  "drift-detection-details":{
    "drifted-stack-resource-count":1
  },
  "client-request-token":""
}
}

```

StackSet Détails de l'événement de changement de statut

Vous trouverez ci-dessous les champs détaillés relatifs aux événements StackSet de changement de statut.

Les `detail-type` champs source et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```

{
  . . . ,
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "status-details": {
      "status":"string"
    }
  }
}

```

detail-type

Identifie le type d'événement.

Pour les événements d' StackSet état, cette valeur est `CloudFormation StackSet Status Change`.

source

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements d' `StackSet` état, ces données incluent :

`stack-set-arn`

Amazon Resource Name (ARN) associé à l'ensemble de piles.

`status-details`

`status`

Le `StackSet` statut.

Valeurs valides : `ACTIVE` | `DELETED`

Exemple Exemple : événement `StackSet` de changement de statut

Voici un exemple d'événement de changement de `StackSet` statut. Cet événement indique que CloudFormation a supprimé l'ensemble de piles spécifié.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-23T17:06:18Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663"
  ],
  "detail": {
    "stack-set-arn" : "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663",
    "status-details": {
```

```
        "status": "DELETED"
    }
}
```

StackSet Stack Instance Status Modifier le détail de l'événement

Vous trouverez ci-dessous les champs détaillés des événements relatifs au statut des instances de StackSet pile.

Les `detail-type` champs source et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```
{
  . . . ,
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-id" : "string",
    "action" : "string",
    "status-details": {
      "status": "string",
      "status-reason": "string",
      "detailed-status": "string"
    }
  }
}
```

detail-type

Identifie le type d'événement.

Pour les événements relatifs au statut des instances de StackSet pile, cette valeur est `CloudFormation StackSet StackInstance Status Change`.

source

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements relatifs au statut des instances de StackSet stack, ces données incluent :

`stack-set-arn`

Le nom de ressource Amazon (ARN) associé au StackSet.

`stack-id`

ID de pile unique associé à l'instance de pile.

`action`

Type d'opération d'ensemble de pile.

Valeurs valides : CREATE | UPDATE | DELETE | DETECT_DRIFT

`status-details`

`status`

État de l' StackSet instance.

Pour en savoir plus, consultez [Codes d'état de l'instance de pile](#).

Valeurs valides : CURRENT | OUTDATED | INOPERABLE

`status-reason`

Raison du statut de l' StackSet instance.

`detailed-status`

État détaillé de l' StackSet instance.

Valeurs valides : CANCELLED | FAILED | FAILED_IMPORT | INOPERABLE | PENDING | RUNNING | SKIPPED_SUSPENDED_ACCOUNT | SUCCEEDED

Exemple Exemple : événement de modification du statut d'une instance StackSet Stack

Voici un exemple d'événement StackSet Stack Instance Stack Status Change.

```
{
```

```

"version": "0",
"id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
"detail-type": "CloudFormation StackSet StackInstance Status Change",
"source": "aws.cloudformation",
"account": "111122223333",
"time": "2021-09-22T19:19:23Z",
"region": "us-east-1",
"resources": [
  "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-
d041-44ad-94f8-b8268aca1e59"
],
"detail": {
  "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/
test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
  "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
  "status-details": {
    "status": "OUTDATED",
    "status-reason": "User Initiated",
    "detailed-status": "PENDING"
  }
}
}

```

StackSet État de l'opération Modifier le détail de l'événement

Vous trouverez ci-dessous les champs détaillés relatifs aux événements de changement de statut de l' StackSet opération.

Les detail-type champs source et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```

{
  . . . ,
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-set-operation-id" : "string",
    "status-details": {
      "status": "string"
    }
  }
}

```

```
}  
}
```

detail-type

Identifie le type d'événement.

Pour les événements relatifs à l'état des StackSet opérations, cette valeur est `CloudFormation StackSet Operation Status Change`.

source

Identifie le service qui a généré l'événement. Pour les CloudFormation événements, cette valeur est `aws.cloudformation`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements relatifs à l'état du StackSet fonctionnement, ces données incluent :

stack-set-arn

Le nom de ressource Amazon (ARN) associé au StackSet.

stack-set-operation-id

L'identifiant unique associé à l' StackSet opération.

status-details

status

État de StackSet l'opération.

Pour en savoir plus, consultez [StackSets codes de statut](#).

Valeurs Valides : `RUNNING` | `SUCCEEDED` | `FAILED` | `STOPPING` | `STOPPED` | `QUEUED`

Exemple Exemple : événement de modification du statut de l' StackSet opération

Voici un exemple d'événement de changement de statut d' StackSet opération. Les détails de l'événement qui CloudFormation a effectué avec succès l'opération demandée sur l'ensemble de piles spécifié.

```
{
  "version": "0",
  "id": "4de89905-fd92-6a6b-9509-23c04bcb6a21",
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T05:46:24Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59"
  ],
  "detail": {
    "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
    "stack-set-operation-id": "ce69adce-2221-4483-8c4b-c51f284f25e8",
    "status-details": {
      "status": "SUCCEEDED"
    }
  }
}
```

État de synchronisation du référentiel Modifier le détail de l'événement

Vous trouverez ci-dessous les champs détaillés pour les événements de modification de l'état de synchronisation du référentiel.

Les `detail-type` champs `source` et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```
{
  . . . ,
  "detail-type": "Git Sync Repository Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
    "connectionArn" : "string",
    "providerType" : "string",
    "repositoryName": "string",
    "providerType": "string",
    "repositoryName": "string",
    "repositoryArn": "string",
  }
}
```

```
"repositoryLinkId": "string",
"ownerId": "string",
"commit": "string",
"branch": "string",
"syncType": "string",
"status": "string",
"previousSync": "string"
}
}
```

detail-type

Identifie le type d'événement.

Pour les événements relatifs à l'état de synchronisation du référentiel, cette valeur est `Git Sync Repository Sync Status Change`.

source

Identifie le service qui a généré l'événement. Pour les événements de synchronisation Git, cette valeur est `aws.codeconnections`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements d'état de synchronisation du référentiel, ces données incluent :

connectionArn

Le nom de ressource Amazon (ARN) associé à CodeConnections.

providerType

Le fournisseur Git connecté à CloudFormation.

Valeurs valides : `GitHub | GitHub Enterprise | GitLab | BitBucket`

repositoryName

Le nom du référentiel Git.

repositoryArn

L'ARN associé au référentiel Git.

repositoryLinkId

L'identifiant unique associé au lien du référentiel.

ownerId

L'identifiant unique associé au propriétaire du référentiel.

commit

Identifiant unique associé à la validation du référentiel.

branch

ID unique associé à la branche du référentiel.

syncType

Type de synchronisation en cours d'exécution.

status

État actuel de synchronisation du référentiel.

Valeurs valides : FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

État de synchronisation antérieur à l'état actuel.

Valeurs valides : FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Exemple Exemple : événement de modification de l'état de synchronisation du référentiel

Voici un exemple d'événement de changement d'état de référentiel de synchronisation. Les détails de l'événement qui CodeConnections a correctement synchronisé le référentiel.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "GitSync Repository Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
```

```

"resources": [{"arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000"},],
"detail": {
  "connectionArn": "arn:aws:codestar-connections:us-east-1:111122223333:connection/sample-connection-id",
  "providerType": "GitHub",
  "repositoryName": "sample-repository-name",
  "repositoryArn": "arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000",
  "repositoryLinkId": "550e8400-e29b-41d4-a716-446655440000",
  "ownerId": "sample-owner-id",
  "commit": "sample-commit-id",
  "branch": "main",
  "syncType": "CFN_STACK_SYNC",
  "status": "SUCCEEDED",
  "previousStatus": "IN_PROGRESS",
}
}

```

Détails de l'événement « Modification de l'état de synchronisation des ressources »

Vous trouverez ci-dessous les champs détaillés pour les événements de modification de l'état de synchronisation des ressources.

Les `detail-type` champs source et sont inclus car ils contiennent des valeurs spécifiques pour les événements.

```

{
  . . . ,
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
    "providerType" : "string",
    "commit" : "string",
    "repositoryName": "string",
    "branch": "string",
    "syncType": "string",
    "syncTarget": "string",
    "status": "string",
    "previousSync": "string"
  }
}

```

```
}  
}
```

detail-type

Identifie le type d'événement.

Pour les événements relatifs à l'état de synchronisation du référentiel, cette valeur est `Git Sync Repository Sync Status Change`.

source

Identifie le service qui a généré l'événement. Pour les événements de synchronisation Git, cette valeur est `aws.codeconnections`.

detail

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Pour les événements liés à l'état de synchronisation des ressources, ces données comprennent :

providerType

Le fournisseur Git connecté à CloudFormation.

Valeurs valides : `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

commit

Identifiant unique associé à la validation du référentiel.

repositoryName

Le nom du référentiel Git.

branch

ID unique associé à la branche du référentiel.

syncType

Type de synchronisation en cours d'exécution.

syncTarget

La pile cible pour la synchronisation des ressources.

status

État actuel de synchronisation du référentiel.

Valeurs valides : FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

État de synchronisation antérieur à l'état actuel.

Valeurs valides : FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Exemple Exemple : événement de modification du statut de synchronisation des ressources

Voici un exemple d'événement de modification de l'état de synchronisation des ressources. Les détails de l'événement qui CodeConnections a correctement synchronisé la ressource.

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:cloudformation:us-east-1:111122223333:stack/targetStack1"],
  "detail": {
    "providerType": "GitHub",
    "commit": "sample-commit-id",
    "repositoryName": "sample-repository-name",
    "branch": "main",
    "syncType": "CFN_STACK_SYNC",
    "syncTarget": "arn:aws:aws:cloudformation:us-east-1:111122223333:stack/targetStack1",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS"
  }
}
```

Comprendre les CloudFormation quotas

Vous avez un compte AWS CloudFormation des quotas que vous devrez peut-être connaître lors de la création de modèles et de la création de piles. Vous éviterez ainsi les erreurs liées à ces quotas et n'aurez pas à remanier les modèles ou les piles.

Le tableau suivant indique les CloudFormation quotas.

Quotas	Description	Value	Stratégie d'ajustement
Données de condition d'attente cfn-signal	La quantité maximale de données que <code>cfn-signal</code> peut transmettre.	4,096 bytes	Pour transmettre une quantité plus importante, envoyez les données dans un compartiment Amazon S3, puis utilisez <code>cfn-signal</code> pour transmettre l'URL Amazon S3 vers ce compartiment.
Réponse de ressource personnalisée	La quantité maximale de données qu'un fournisseur de ressource personnalisée peut transmettre.	4,096 bytes	
Références dynamiques par modèle	Le nombre maximum de références dynamiques autorisées dans un modèle de CloudFormation unique.	60 références dynamiques dans un modèle de pile	
Hooks par compte	Nombre maximal de hooks autorisés par compte et par région.	100 hooks	

Quotas	Description	Value	Stratégie d'ajustement
Hooks par ressource	Nombre maximal de hooks pouvant être configurés par ressource.	100 hooks	
Taille de configuration d'un hook	Quantité maximale de données de configuration qu'un Hook peut stocker.	204,8 Ko	
Mappages	Le nombre maximum de mappages que vous pouvez déclarer dans votre CloudFormation modèle.	200 mappages	Pour spécifier davantage de mappages, divisez le modèle en plusieurs modèles en utilisant , par exemple, des piles imbriquées .
Attributs de mappage	Nombre maximal d'attributs de mappage pour chaque mappage que vous pouvez déclarer dans votre CloudFormation modèle.	200 attributs	Pour spécifier davantage d'attributs de mappage, divisez les attributs en plusieurs mappages.
Nom du mappage et nom de l'attribut de mappage	Taille maximale de chaque nom de mappage.	255 caractères	
Modules	Le nombre maximum de modules que vous pouvez enregistrer dans le CloudFormation registre, par compte et par région.	100 modules	

Quotas	Description	Value	Stratégie d'ajustement
Versions de modules	Le nombre maximum de versions que vous pouvez enregistrer dans le CloudFormation registre pour un module donné.	100 versions	Pour enregistrer de nouvelles versions, commencez par DeregisterType désenregistrer les versions que vous n'utilisez plus.
Piles imbriquées	Nombre maximal de CloudFormation ressources qu'une pile imbriquée peut créer, mettre à jour ou supprimer par opération. Par exemple, vous pouvez avoir une hiérarchie de piles imbriquées contenant plus de 2 500 ressources au total, mais vous ne pouvez pas créer, mettre à jour ou supprimer plus de 2 500 de ces ressources dans un seul déploiement.	2 500	Divisez la hiérarchie des piles en différentes piles.
Sorties	Le nombre maximum de sorties que vous pouvez déclarer dans votre CloudFormation modèle.	200 sorties	

Quotas	Description	Value	Stratégie d'ajustement
Nom de sortie	Taille maximale d'un nom de sortie.	255 caractères	
Paramètres	Le nombre maximum de paramètres que vous pouvez déclarer dans votre CloudFormation modèle.	200 paramètres	Pour spécifier davantage de paramètres, vous pouvez utiliser des mappages ou des listes afin d'attribuer plusieurs valeurs à un seul paramètre.
Nom du paramètre	Taille maximale d'un nom de paramètre.	255 caractères	
Valeur de paramètre	Taille maximale d'une valeur de paramètre.	4,096 bytes	Pour utiliser une valeur paramétrée plus grande, créez plusieurs paramètres, puis utilisez la fonction Fn::Join pour concaténer plusieurs valeurs en une seule.
Ressources privées	Le nombre maximum de ressources privées que vous pouvez enregistrer dans le CloudFormation registre par compte et par région.	50 ressources privées	

Quotas	Description	Value	Stratégie d'ajustement
Versions de ressources privées	Nombre maximal de versions que vous pouvez enregistrer dans le CloudFormation registre pour une ressource privée donnée.	50 ressources privées	Pour enregistrer de nouvelles versions, commencez par DeregisterType désenregistrer les versions que vous n'utilisez plus.
Ressources	Le nombre maximum de ressources que vous pouvez déclarer dans votre CloudFormation modèle.	500 ressources	Pour spécifier davantage de ressources, divisez le modèle en plusieurs modèles en utilisant , par exemple, des piles imbriquées .
Ressources dans les opérations de pile simultanées	Nombre maximal de ressources pouvant être impliquées dans des opérations de pile (création, mise à jour ou suppression d'opérations) dans votre région à un moment donné.	Utilisez l' DescribeAccountLimits API pour déterminer la limite actuelle d'un compte dans une région spécifique.	
Nom de la ressource	Taille maximale d'un nom de ressource.	255 caractères	

Quotas	Description	Value	Stratégie d'ajustement
Piles	Le nombre maximum de CloudFormation piles que vous pouvez créer.	2 000 piles	Pour créer plus de piles, supprimez les piles dont vous n'avez plus besoin ou demandez une augmentation de la limite maximale de piles dans votre Compte AWS. Pour plus d'informations, consultez la section Quotas du service AWS dans le Références générales AWS .
Nom de la pile	Taille maximale autorisée pour un nom de pile.	128 caractères	

Quotas	Description	Value	Stratégie d'ajustement
StackSets	Le nombre maximum d'ensembles de CloudFormation piles que vous pouvez créer dans votre compte administrateur.	1000 ensembles de piles	Pour créer davantage d'ensembles de piles, supprimez ceux dont vous n'avez plus besoin ou demandez une augmentation de la limite maximale d'ensembles de piles dans votre Compte AWS. Pour plus d'informations, consultez la section Quotas du service AWS dans le Références générales AWS.
Instances de piles	Nombre maximal d'instances de pile que vous pouvez créer par ensemble de piles.	100 000 instances de piles par ensemble de piles	Pour créer davantage d'instances, supprimez celles dont vous n'avez plus besoin ou demandez une augmentation de la limite maximale d'instances dans votre Compte AWS. Pour plus d'informations, consultez la section Quotas du service AWS dans le Références générales AWS.

Quotas	Description	Value	Stratégie d'ajustement
StackSets opérations d'instance	Nombre maximal d'instances de pile, tous ensembles confondus, sur lesquelles vous pouvez exécuter des opérations simultanément dans chaque région, par compte administrateur.	10 000 opérations	Cette limite s'applique à tous les ensembles de piles impliqués dans une région. Elle inclut les instances de pile affectées par les opérations de création et de mise à jour des ensembles de piles, ainsi que la création, la mise à jour ou la suppression directes d'instances de pile.
StackSets opérations en file d'attente	Nombre maximal d'opérations pouvant être mises en file d'attente pour un ensemble de piles à un moment donné.	10 000 opérations	
Piles importées à l'aide d'un objet S3 par opération d'ensemble de piles	Nombre maximal de piles que vous pouvez importer à l'aide d'un objet S3 dans le cadre d'une seule opération d'ensemble de piles.	200 piles	
Piles importées à l'aide d'identifiants de pile intégrés par opération d'ensemble de piles	Le nombre maximum de piles que vous pouvez importer à l'aide d'une pile intégrée IDs par opération d'ensemble de piles.	10 piles	

Quotas	Description	Value	Stratégie d'ajustement
Taille du corps du modèle dans une demande	Taille maximale du corps du modèle que vous pouvez transmettre dans une requête <code>CreateStack</code> , <code>UpdateStack</code> ou <code>ValidateTemplate</code> .	51,200 bytes	Pour utiliser un corps de modèle plus grand, divisez le modèle en plusieurs modèles en utilisant, par exemple, des piles imbriquées . Vous pouvez également importer le modèle dans un compartiment Amazon S3.
Taille du corps du modèle dans un objet Amazon S3	Taille maximale du corps de modèle que vous pouvez transmettre dans un objet Amazon S3 pour une requête <code>CreateStack</code> , <code>UpdateStack</code> , <code>ValidateTemplate</code> avec une URL de modèle Amazon S3.	1 Mo	Pour utiliser un corps de modèle plus grand, divisez le modèle en plusieurs modèles en utilisant, par exemple, des piles imbriquées . Vous pouvez également utiliser la minification pour réduire la taille du CloudFormation modèle.
Description du modèle	Taille maximale autorisée pour la description d'un modèle.	1,024 bytes	
Versions par crochet	Nombre maximal de versions que vous pouvez créer par hook.	100 versions	

Disponibilité des fonctions

Certaines fonctionnalités de ne sont CloudFormation peut-être pas disponibles dans toutes les régions. Pour plus d'informations sur Régions AWS, consultez le [Tableau des régions d'infrastructure mondiale](#).

- Les [macros](#) ne sont actuellement pas disponibles dans la région suivante :
 - Asie-Pacifique (Jakarta)
- L'[exécution de blue/green déploiements ECS par le biais de CodeDeploy l'utilisation](#) n'CloudFormationest actuellement pas disponible dans les régions suivantes :
 - Afrique (Le Cap)
 - Asie-Pacifique (Osaka)
 - Europe (Milan)

StackSets et macros

StackSets ne prend actuellement pas en charge la création ou la mise à jour d'ensembles de piles avec des autorisations gérées par les services à partir de modèles contenant des macros. Cela inclut les transformations, qui sont des macros hébergées par CloudFormation. Pour plus d'informations sur les macros, consultez [Macros de modèle](#).

Résolution des problèmes CloudFormation

Lors de l'utilisation CloudFormation, vous pouvez rencontrer des problèmes lors de la création, de la mise à jour ou de la suppression de CloudFormation piles. Les sections suivantes peuvent vous aider à résoudre les problèmes courants qui peuvent se poser.

Pour des questions générales sur CloudFormation, consultez le [CloudFormation FAQs](#). Vous pouvez également rechercher des réponses et poser des questions dans la [CloudFormation communauté](#) sur AWS re:Post.

Rubriques

- [Guide de dépannage](#)
- [Résolution des erreurs](#)
- [Contacter l'assistance](#)

Guide de dépannage

Si CloudFormation vous ne parvenez pas à créer, mettre à jour ou supprimer votre pile, vous pouvez consulter les messages d'erreur ou les journaux pour en savoir plus sur le problème. Les tâches suivantes décrivent les méthodes générales de résolution d'un CloudFormation problème. Pour plus d'informations sur des erreurs spécifiques et les solutions, consultez [Résolution des erreurs](#).

- Utilisez le filtrage IDs des opérations et des défaillances pour un dépannage ciblé. CloudFormation attribue une opération unique IDs aux événements de déploiement, ce qui facilite le suivi et la corrélation des événements connexes. Vous pouvez filtrer les événements de pile par ID d'opération pour vous concentrer sur des phases de déploiement spécifiques et utiliser des filtres d'échec pour identifier rapidement la cause première des échecs de déploiement. Consultez [Afficher les événements de la pile par opération](#)
- Utilisez la CloudFormation console pour consulter l'état de votre stack. Cette console vous permet de voir une liste des événements liés à une pile pendant que cette dernière est créée, mise à jour ou supprimée. Les événements de pile incluent des opérations IDs qui vous aident à corréler les événements connexes et à suivre les opérations individuelles. Vous pouvez cliquer sur l'ID de l'opération pour un dépannage ciblé. Dans la liste des événements d'opération, recherchez l'événement de défaillance, puis visualisez le statut et la raison de cet événement. La raison du statut peut contenir un message d'erreur provenant CloudFormation ou provenant d'un service particulier qui peut vous aider à résoudre votre problème. Pour plus d'informations sur l'affichage

des événements de pile avec une opération IDs, consultez [Afficher les événements de la pile par opération](#).

- Pour les problèmes liés à Amazon EC2, consultez les journaux `cloud-init` et `cfn`. Ces journaux sont publiés dans le répertoire `/var/log/` de l'instance Amazon EC2. Ces journaux capturent les processus et les sorties de commandes lors CloudFormation de la configuration de votre instance. Pour Windows, consultez les sections EC2 Configurer le service dans `%ProgramFiles%\Amazon\EC2ConfigService`, EC2 Launch dans `%ProgramData%\Amazon\EC2-Windows\Launch\Logs`, EC2 Launch v2 dans `%ProgramData%\Amazon\EC2Launch\log` et `cfn` se connecter. `C:\cfn\log`

Vous pouvez également configurer votre CloudFormation modèle pour que les journaux soient publiés sur Amazon CloudWatch, qui affiche les journaux dans le AWS Management Console afin que vous n'ayez pas à vous connecter à votre instance Amazon EC2. Pour plus d'informations, voir [Afficher CloudFormation les journaux dans la console](#) dans le blog de gestion des applications.

Résolution des erreurs

Lorsque vous rencontrez les erreurs suivantes dans votre CloudFormation pile, vous pouvez utiliser les solutions suivantes pour vous aider à trouver la source des problèmes et à les résoudre.

Rubriques

- [Échec de la suppression de la pile](#)
- [Erreur de dépendance](#)
- [AWS Config et AWS Systems Manager conflits](#)
- [Erreur d'analyse d'un paramètre lors de la transmission d'une liste](#)
- [Autorisations IAM insuffisantes](#)
- [Valeur non valide ou propriété de ressource non prise en charge](#)
- [Quota dépassé](#)
- [Les piles imbriquées restent bloquées à l'état UPDATE_COMPLETE_CLEANUP_IN_PROGRESS, UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS ou UPDATE_ROLLBACK_IN_PROGRESS](#)
- [Aucune mise à jour à effectuer](#)
- [La ressource n'a pas pu se stabiliser lors d'une opération de création, de mise à jour ou de suppression de pile](#)

- [Le groupe de sécurité n'existe pas dans le VPC](#)
- [Échec de la restauration de la mise à jour](#)
- [La condition d'attente n'a pas reçu le nombre de signaux requis à partir d'une instance Amazon EC2](#)
- [Ressource retirée de la pile mais non supprimée](#)

Échec de la suppression de la pile

Pour résoudre ce problème, essayez la méthode suivante :

- Certaines ressources doivent être vides avant de pouvoir être supprimées. Par exemple, vous devez supprimer tous les objets d'un compartiment Amazon S3 ou supprimer toutes les instances d'un groupe de sécurité Amazon EC2 avant de pouvoir supprimer ce compartiment ou ce groupe de sécurité.
- Vérifiez que vous disposez des autorisations IAM nécessaires pour supprimer les ressources de la pile. Outre CloudFormation les autorisations, vous devez être autorisé à utiliser les services sous-jacents, tels qu'Amazon S3 ou Amazon EC2.
- Lorsque les piles sont dans l'`DELETE_FAILED` état où elles n'ont pas CloudFormation pu supprimer une ressource, réexécutez la [suppression](#) avec le `RetainResources` paramètre et spécifiez la ressource qui ne CloudFormation peut pas être supprimée. CloudFormation supprime la pile sans supprimer la ressource conservée. La conservation des ressources s'avère utile lorsque vous ne pouvez pas supprimer une ressource, telle qu'un compartiment S3 qui contient les objets que vous souhaitez conserver, mais que vous voulez supprimer la pile.

Après avoir supprimé la pile, vous pouvez supprimer manuellement les ressources conservées à l'aide du AWS service associé.

Vous pouvez également envisager d'utiliser l'option `FORCE_DELETE_STACK` avec le paramètre `DeletionMode`. Pour plus d'informations sur la suppression forcée d'une pile, consultez [Supprimer une pile de la CloudFormation console](#).

- Vous ne pouvez pas supprimer des piles pour lesquelles la protection contre la résiliation est activée. Si vous essayez de supprimer une pile pour laquelle la protection contre la résiliation est activée, la suppression échoue et la pile, et son statut, restent inchangés. Désactivez la protection contre la résiliation sur la pile, puis effectuez à nouveau l'opération de suppression.

Cela concerne également les [piles imbriquées](#) dont les piles racine sont protégées contre la résiliation. Désactivez la protection contre la résiliation sur la pile racine, puis effectuez à nouveau l'opération de suppression. Il est fortement recommandé de ne pas supprimer les piles imbriquées directement, mais de les supprimer uniquement dans le cadre de la suppression de la pile racine et de l'ensemble de ses ressources.

Pour de plus amples informations, veuillez consulter [CloudFormation Empêcher les piles d'être supprimées](#).

- Pour tous les autres problèmes, si vous en avez AWS Support, vous pouvez créer un Support dossier. Consultez [Contacter l'assistance](#).

Erreur de dépendance

Pour résoudre une erreur de dépendance, ajoutez un attribut `DependsOn` aux ressources qui dépendent d'autres ressources dans votre modèle. Dans certains cas, vous devez déclarer explicitement les dépendances afin de CloudFormation pouvoir créer ou supprimer des ressources dans le bon ordre. Par exemple, si vous créez une adresse IP Elastic et un VPC avec une passerelle Internet dans la même pile, l'adresse IP Elastic dépend de l'association à cette passerelle Internet. Pour plus d'informations, voir la section consacrée à l'[Attribut DependsOn](#).

AWS Config et AWS Systems Manager conflits

AWS Config et AWS Systems Manager peut automatiser les tâches de gestion de l'infrastructure susceptibles d'entraîner des conflits lors du déploiement d'une CloudFormation pile. Procédez comme suit pour éviter tout conflit potentiel :

- Vérifiez la configuration de AWS Config et de Systems Manager dans le fichier associé Compte AWS et Région AWS.
- Vérifiez les règles actives ou les documents d'automatisation susceptibles d'être déclenchés lors d'un CloudFormation déploiement. Ceux-ci peuvent potentiellement causer des conflits ou des dépendances de ressources qui entrent en conflit avec votre déploiement.
- Vérifiez que votre CloudFormation modèle contient toutes les ressources gérées par AWS Config et Systems Manager. Vérifiez les chevauchements ou interdépendances potentiels et envisagez d'ajuster le modèle ou la configuration de l'automatisation pour éviter les conflits.
- Désactivez ou suspendez temporairement AWS Config les règles associées ou les automatisations de Systems Manager pendant CloudFormation le déploiement. N'oubliez pas de restaurer

les configurations d'origine après le déploiement réussi afin de maintenir le niveau souhaité d'automatisation et de conformité.

- Consultez les CloudFormation journaux et les messages d'erreur pour détecter toute référence à AWS Config des problèmes liés à Systems Manager afin d'identifier la source du conflit.

Pour plus d'informations sur AWS Config les règles, consultez la section [Évaluation des ressources avec AWS Config Rules](#).

Pour plus d'informations sur les automatisations Systems Manager, consultez [AWS Systems Manager Automation](#).

Erreur d'analyse d'un paramètre lors de la transmission d'une liste

Lorsque vous utilisez le AWS CLI ou CloudFormation pour transmettre une liste, ajoutez le caractère d'échappement (\) avant chaque virgule. L'exemple suivant montre comment spécifier un paramètre d'entrée avec l' AWS CLI.

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

Autorisations IAM insuffisantes

Lorsque vous travaillez avec une CloudFormation pile, vous avez non seulement besoin d'autorisations pour l'utiliser CloudFormation, mais vous devez également avoir l'autorisation d'utiliser les services sous-jacents décrits dans votre modèle. Par exemple, si vous créez un compartiment Amazon S3 ou si vous lancez une instance Amazon EC2, vous devez être autorisé à accéder à Amazon S3 ou à Amazon EC2. Passez en revue votre politique IAM et vérifiez que vous disposez des autorisations nécessaires avant d'utiliser les piles CloudFormation . Pour plus d'informations, voir [Contrôlez CloudFormation l'accès avec Gestion des identités et des accès AWS](#).

Valeur non valide ou propriété de ressource non prise en charge

Lorsque vous créez ou mettez à jour une CloudFormation pile, celle-ci peut échouer en raison de paramètres d'entrée non valides, de noms de propriétés de ressources non pris en charge ou de valeurs de propriétés de ressource non prises en charge. Pour les paramètres d'entrée, vérifiez que la ressource existe. Par exemple, lorsque vous spécifiez une paire de clés Amazon EC2 ou un ID de VPC, la ressource doit exister dans votre compte et dans la région dans laquelle vous créez ou mettez à jour la pile. Pour vous assurer que les valeurs sont valides, vous pouvez utiliser les [types de paramètres spécifiques àAWS](#).

Pour les noms et les valeurs des propriétés de ressource, mettez à jour le modèle pour qu'il utilise des noms et des valeurs valides. Pour plus d'informations, veuillez consulter la [référence des types de ressources et de propriétés AWS](#).

Quota dépassé

Vérifiez que vous n'avez pas atteint un quota de ressource. Par exemple, le nombre maximal d'instances à la demande Amazon EC2 que vous pouvez lancer par défaut est de 5. Si vous essayez de créer plus d'instances à la demande Amazon EC2 que le quota défini dans votre compte, l'opération échoue et vous recevez l'erreur `Status=start_failed`. Pour afficher les quotas AWS par défaut et par service, veuillez consulter la rubrique [AWS service quotas](#) dans Références générales AWS.

Pour les CloudFormation quotas et les stratégies d'ajustement, voir. [Comprendre les CloudFormation quotas](#)

De même, lors d'une mise à jour, si une ressource est remplacée, une nouvelle ressource est CloudFormation créée avant de supprimer l'ancienne. Ce remplacement pourrait entraîner le dépassement du quota de ressources associé à votre compte, ce qui provoquerait l'échec de la mise à jour. Vous pouvez supprimer des ressources inutilisées ou demander une [augmentation de quota](#).

Les piles imbriquées restent bloquées à l'état

UPDATE_COMPLETE_CLEANUP_IN_PROGRESS,
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS ou
UPDATE_ROLLBACK_IN_PROGRESS

La restauration d'une pile imbriquée a échoué. En raison des dépendances potentielles entre les piles imbriquées, le nettoyage des ressources des piles imbriquées CloudFormation ne commence pas tant que toutes les piles imbriquées n'ont pas été mises à jour ou annulées. Lorsqu'une pile imbriquée ne parvient pas à revenir en arrière, toutes les opérations sont CloudFormation annulées, quel que soit l'état dans lequel se trouvent les autres piles imbriquées. Une pile imbriquée qui a terminé la mise à jour ou la restauration mais qui n'a pas reçu de signal indiquant de CloudFormation commencer le nettoyage car une autre pile imbriquée n'a pas réussi à revenir en arrière est dans un état `UPDATE_COMPLETE_CLEANUP_IN_PROGRESS` ou `UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS`. Lorsqu'une pile imbriquée dont la mise à jour a échoué ne reçoit pas le signal pour commencer sa restauration, son état indique `UPDATE_ROLLBACK_IN_PROGRESS`.

Une pile imbriquée peut ne pas être annulée en raison de modifications apportées en dehors de CloudFormation, lorsque le modèle de pile ne reflète pas exactement l'état de la pile. Une pile imbriquée peut également échouer si le délai d'expiration du signal pour son groupe Auto Scaling n'était pas suffisant lorsque ce dernier a été créé ou mis à jour.

Pour réparer la pile, contactez [AWS Support](#).

Aucune mise à jour à effectuer

Pour mettre à jour une CloudFormation pile, vous devez soumettre des modifications de modèle ou de valeur de paramètre à CloudFormation. Toutefois, certaines modifications du modèle CloudFormation ne seront pas reconnues comme des mises à jour, telles que les modifications apportées à une politique de suppression, à une politique de mise à jour, à une déclaration de condition ou à une déclaration de sortie. Si vous avez besoin d'effectuer uniquement ces modifications, vous pouvez ajouter ou modifier les attributs de métadonnées de vos ressources. L'attribut de métadonnées peut être n'importe quelle valeur arbitraire, car CloudFormation il n'interprète pas son contenu. Pour plus d'informations, voir la section consacrée à l'[Attribut Metadata](#).

La ressource n'a pas pu se stabiliser lors d'une opération de création, de mise à jour ou de suppression de pile

Une ressource n'a pas répondu car l'opération a dépassé le CloudFormation délai imparti ou parce qu'un AWS service a été interrompu. En cas d'interruption de service, [vérifiez](#) que le AWS service concerné est en cours d'exécution, puis réessayez l'opération de pile.

Si les AWS services fonctionnent correctement, vérifiez si votre pile contient l'une des ressources suivantes :

- `AWS::AutoScaling::AutoScalingGroup` pour les opérations de création, de mise à jour et de suppression
- `AWS::CertificateManager::Certificate` pour les opérations de création
- `AWS::CloudFormation::Stack` pour les opérations de création, de mise à jour et de suppression
- `AWS::ElasticSearch::Domain` pour les opérations de mise à jour
- `AWS::RDS::DBCluster` pour les opérations de création et de mise à jour
- `AWS::RDS::DBInstance` pour les opérations de création, de mise à jour et de suppression
- `AWS::Redshift::Cluster` pour les opérations de mise à jour

Pour ces ressources, les opérations peuvent prendre plus de temps que le délai d'expiration par défaut. Le délai d'expiration varie en fonction de la ressource et des informations d'identification que vous utilisez. Pour allonger le délai d'expiration, spécifiez un [rôle de service](#) au moment d'effectuer l'opération de pile. Si vous utilisez déjà un rôle de service ou si votre pile contient une ressource qui n'est pas répertoriée, contactez le [AWS Support](#).

Si votre pile indique l'état `UPDATE_ROLLBACK_FAILED`, consultez [Échec de la restauration de la mise à jour](#).

Le groupe de sécurité n'existe pas dans le VPC

Vérifiez que le groupe de sécurité se trouve dans le VPC que vous avez spécifié. Si tel est le cas, vérifiez que vous avez spécifié l'ID de groupe de sécurité, et pas le nom du groupe de sécurité. Par exemple, la ressource `AWS::EC2::SecurityGroupIngress` possède les propriétés `SourceSecurityGroupName` et `SourceSecurityGroupId`. Pour les groupes de sécurité d'un VPC, vous devez utiliser la propriété `SourceSecurityGroupId` et définir l'ID du groupe de sécurité.

Échec de la restauration de la mise à jour

Une ressource dépendante ne parvient pas à rétablir son état d'origine, ce qui entraîne l'échec de la restauration (état `UPDATE_ROLLBACK_FAILED`). Par exemple, la restauration a peut-être lieu dans une instance de base de données obsolète qui a été supprimée en dehors de CloudFormation. Comme il CloudFormation ne sait pas que la base de données a été supprimée, il suppose que l'instance de base de données existe toujours et tente d'y revenir, ce qui entraîne l'échec de l'annulation de la mise à jour.

Selon l'origine du problème, vous pouvez corriger manuellement l'erreur et poursuivre la restauration. De cette façon, vous pourrez rétablir l'état de fonctionnement de la pile (état `UPDATE_ROLLBACK_COMPLETE`), puis réessayer de mettre à jour la pile. La liste suivante décrit les solutions à des erreurs courantes qui entraînent l'échec de restauration d'une mise à jour :

- Le nombre de signaux requis n'a pas été reçu

Utilisez la commande [signal-resource](#) pour envoyer manuellement le nombre de signaux de réussite requis à la ressource qui les attend, puis poursuivez la restauration de la mise à jour. Par exemple, pendant la restauration d'une mise à jour, les instances d'un groupe Auto Scaling ne parviennent pas toujours à signaler la réussite dans le délai spécifié. Dans ce cas, envoyez manuellement les signaux de réussite à ce groupe Auto Scaling. Lorsque vous

poursuivez l'annulation de la mise à jour CloudFormation , elle reçoit vos signaux et procède à la restauration.

- Les modifications apportées à une ressource ont été effectuées en dehors de CloudFormation

Synchronisez manuellement les ressources pour qu'elles correspondent au modèle de la pile d'origine, puis poursuivez la restauration de la mise à jour. Par exemple, si vous avez supprimé manuellement une ressource à laquelle CloudFormation vous tentez de revenir, vous devez créer manuellement cette ressource avec le même nom et les mêmes propriétés que dans la pile d'origine.

- Autorisations insuffisantes

Vérifiez que vous disposez des autorisations IAM suffisantes pour modifier des ressources, puis poursuivez la restauration de la mise à jour. Par exemple, votre politique IAM peut vous permettre de créer un compartiment S3, mais pas de le modifier. Ajoutez les actions de modification à votre politique.

- Jeton de sécurité non valide

CloudFormation nécessite un nouveau jeu d'informations d'identification. Aucune modification n'est requise. Continuez à restaurer la mise à jour, ce qui actualisera les informations d'identification.

- Erreur de limitation

Supprimez les ressources dont vous n'avez pas besoin ou demandez une [augmentation de quota](#), puis poursuivez la restauration de la mise à jour. Par exemple, si le quota du nombre d'instances à la demande EC2 s'élève à 5 pour votre compte et si la restauration de la mise à jour dépasse cette valeur, elle échouera.

- La ressource ne s'est pas stabilisée

Une ressource n'a pas répondu car l'opération a peut-être dépassé le CloudFormation délai d'expiration ou parce qu'un AWS service a peut-être été interrompu. Aucune modification n'est requise. Une fois l'opération sur la ressource terminée ou le AWS service rétabli, continuez à annuler la mise à jour.

Pour continuer à annuler une mise à jour, vous pouvez utiliser la CloudFormation console ou l'interface de ligne de AWS commande (AWS CLI). Pour de plus amples informations, veuillez consulter [Poursuite de la restauration d'une mise à jour](#).

Si aucune de ces solutions ne fonctionne, vous pouvez ignorer les ressources qui ne CloudFormation peuvent pas être restaurées avec succès. Pour plus d'informations, consultez le `ResourcesToSkip` paramètre de l'opération [ContinueUpdateRollback](#) d'API dans la référence AWS CloudFormation d'API. CloudFormation définit le statut des ressources spécifiées sur `UPDATE_COMPLETE` et continue de faire reculer la pile. Une fois la restauration terminée, l'état des ressources ignorées ne correspond pas à celui des ressources contenues dans le modèle de pile. Avant de procéder à une autre mise à jour de pile, vous devez modifier les ressources ou mettre à jour la pile de façon à les rendre cohérentes les unes par rapport aux autres. A défaut, les futures mises à jour de la pile risquent d'échouer et votre pile sera irrécupérable.

La condition d'attente n'a pas reçu le nombre de signaux requis à partir d'une instance Amazon EC2

Pour résoudre ce problème, essayez la méthode suivante :

- Assurez-vous que les scripts d' CloudFormation assistance sont installés sur l'AMI que vous utilisez. Si l'AMI n'inclut pas les scripts d'assistant, vous pouvez également les télécharger dans votre instance. Pour plus d'informations, consultez la [référence aux scripts CloudFormation d'assistance](#) dans le Guide de référence des CloudFormation modèles.
- Vérifiez que la commande `cfn-signal` a été correctement exécutée au niveau de l'instance. Vous pouvez consulter des journaux, comme `/var/log/cloud-init.log` ou `/var/log/cfn-init.log`, afin de vous aider à déboguer le lancement de l'instance. Vous pouvez récupérer les journaux en vous connectant à votre instance, mais vous devez [désactiver le rollback en cas d'échec](#), sinon CloudFormation supprimer l'instance lorsque la création de votre pile échoue. Vous pouvez également [publier les journaux](#) sur Amazon CloudWatch. Pour Windows, vous pouvez consulter les connexions `cfn C:\cfn\log` et les connexions au service EC2 Config. `%ProgramFiles%\Amazon\EC2ConfigService`
- Vérifiez que l'instance est connectée à Internet. Si l'instance se trouve dans un VPC, elle doit être pouvoir se connecter à Internet via un périphérique NAT (si elle se trouve dans un sous-réseau privé) ou via une passerelle Internet (s'il s'agit d'un sous-réseau public). Pour tester la connexion Internet de l'instance, essayez d'accéder à une page web publique, tel que `http://aws.amazon.com`. Par exemple, vous pouvez exécuter la commande suivante au niveau de l'instance. Elle doit retourner un code d'état HTTP 200.

```
curl -I https://aws.amazon.com
```


Pour obtenir des informations sur la configuration d'un périphérique NAT, consultez [NAT](#) dans le Amazon VPC Guide de l'utilisateur.

Ressource retirée de la pile mais non supprimée

Lors d'une mise à jour de pile, CloudFormation a supprimé une ressource d'une pile mais ne l'a pas supprimée. La ressource existe toujours mais n'est plus accessible via CloudFormation. Cela peut se produire lors des mises à jour de pile où :

- CloudFormation doit remplacer une ressource existante. Il crée donc d'abord une nouvelle ressource, puis tente de supprimer l'ancienne ressource.
- Vous avez retiré la ressource du modèle de pile, de sorte que CloudFormation tente de supprimer la ressource de la pile.

Cependant, il peut arriver que CloudFormation vous ne puissiez pas supprimer la ressource. Par exemple, si l'utilisateur ne dispose pas des autorisations nécessaires pour supprimer une ressource d'un type donné.

CloudFormation tente de supprimer l'ancienne ressource à trois reprises. S'il n'est pas possible de supprimer l'ancienne ressource, elle est supprimée de la pile et continue de mettre à jour la pile. Lorsque la mise à jour de la pile est terminée, CloudFormation un événement de UPDATE_COMPLETE pile est émis, mais inclut un événement indiquant StatusReason qu'une ou plusieurs ressources n'ont pas pu être supprimées. CloudFormation émet également un DELETE_FAILED événement pour la ressource spécifique, avec un correspondant StatusReason fournissant plus de détails sur les raisons de l' CloudFormation échec de la suppression de la ressource.

Pour résoudre ce problème, supprimez la ressource directement à l'aide de la console ou de l'API pour le service sous-jacent.

Contacter l'assistance

Si c'est le cas AWS Support, vous pouvez créer un dossier de support technique à la <https://console.aws.amazon.com/support/maison#/>. Avant de contacter l'assistance, collectez les informations suivantes :

- ID de la pile. Vous trouverez l'ID de la pile dans l'onglet Présentation de la [console CloudFormation](#). Pour de plus amples informations, veuillez consulter [Afficher les informations relatives à la pile depuis la CloudFormation console](#).

 Important

Ne modifiez pas les ressources de la pile en dehors de CloudFormation. En modifiant les ressources de votre pile à l'extérieur, vous CloudFormation risquez de mettre votre pile dans un état irrécupérable.

- Messages d'erreur associés à la pile. Pour découvrir comment accéder aux messages d'erreur de la pile, consultez [Guide de dépannage](#).
- Pour les problèmes liés à Amazon EC2, veuillez rassembler les journaux `cloud-init` et `cfn`. Ces journaux sont publiés dans le répertoire `/var/log/` de l'instance Amazon EC2. Ces journaux capturent les processus et les sorties de commande lors de la configuration de l'instance. Pour Windows, rassemblez le service EC2 Configure et `cfn` se connecte `%ProgramFiles%\Amazon\EC2ConfigService` et `C:\cfn\log`.

Vous pouvez également rechercher des réponses et poser des questions dans la [CloudFormation communauté](#) sur AWS re:Post.

Résoudre les problèmes liés à l'échec des déploiements de CloudFormation stack avec Amazon Q Developer

Amazon Q Developer est un assistant conversationnel basé sur l'intelligence artificielle générative (IA) qui peut vous aider à comprendre, créer, étendre et exploiter AWS des applications. Pour plus d'informations sur Amazon Q Developer, consultez le [Guide de l'utilisateur Amazon Q Developer](#).

Ce document explique comment utiliser Amazon Q Developer pour vous aider à résoudre les problèmes liés aux CloudFormation déploiements infructueux. Cette caractéristique est conçue pour vous aider à identifier rapidement les problèmes rencontrés lors du déploiement d'une pile et à les résoudre.

Optimisé par Amazon Bedrock : Amazon Q est basé sur Amazon Bedrock et inclut la [détection automatique des abus](#) mise en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'IA.

Rubriques

- [Caractéristiques](#)
- [Comment ça marche](#)
- [Tarification](#)
- [Disponibilité par région](#)
- [Limitations](#)

Caractéristiques

La solution de résolution des problèmes de déploiement des piles avec Amazon Q Developer offre les caractéristiques suivantes :

- Analyse des erreurs — Fournit une easy-to-understand analyse des erreurs liées aux déploiements de CloudFormation stack.
- Suggestions de résolution : fournit des step-by-step instructions pour résoudre les problèmes identifiés.

Comment ça marche

Lorsqu'une création ou une mise à jour d'une CloudFormation pile échoue, un message d'erreur s'affiche dans le AWS Management Console.

Vous pouvez cliquer sur le bouton Diagnostiquer avec Q situé à côté du message d'erreur.

Amazon Q Developer analyse alors l'erreur et fournit une explication du problème.

Pour obtenir une aide supplémentaire, sélectionnez M'aider à résoudre pour recevoir des étapes de résolution détaillées.

Tarification

Pour les informations tarifaires, consultez [Tarification Amazon Q Developer](#).

Disponibilité par région

Amazon Q Developer est disponible dans la plupart des cas Régions AWS. Pour obtenir la liste complète des Régions où Amazon Q Developer est disponible, consultez [Régions prises en charge pour Amazon Q Developer](#) dans le Guide de l'utilisateur Amazon Q Developer.

Limitations

- Cette caractéristique est accessible uniquement depuis la AWS Management Console.
- Cette caractéristique couvre les 20 erreurs les plus courantes. Elle ne couvre pas tous les types d'erreurs possibles.
- Les cas de dépannage complexes peuvent tout de même nécessiter Support une intervention.

Historique du document pour le guide de AWS CloudFormation l'utilisateur

Le tableau suivant décrit les modifications importantes apportées au contenu du guide de AWS CloudFormation l'utilisateur après mai 2018. Vous pouvez vous abonner au flux RSS pour recevoir des notifications sur les mises à jour de la documentation.

Important

Les lignes du tableau décrivant les mises à jour du contenu de référence des modèles à partir de mai 2018 ont été déplacées vers le nouveau Guide de référence des modèles AWS CloudFormation . Pour ces mises à jour et les modifications futures, consultez [l'historique du document pour la référence du AWS CloudFormation modèle](#) dans la référence du AWS CloudFormation modèle.

Modification	Description	Date
Guide PDF disponible	Vous pouvez maintenant télécharger le Guide de l'utilisateur AWS CloudFormation au format PDF.	30 mai 2025
Déplacement du contenu du guide de référence des modèles vers un nouveau guide	CloudFormation a publié le guide CloudFormation de référence des modèles. Pour plus de détails, consultez le guide CloudFormation de référence des modèles .	30 mai 2025
Générateur IaC avec prise en charge de l'analyse partielle	Vous pouvez désormais choisir des types de ressources spécifiques à rechercher, ce qui facilite la génération de modèles Infrastructure-as-Code (IaC) à partir de	27 mars 2025

vos ressources existante
s. Pour plus d'informations,
voir [Lancer une analyse des
ressources avec le générateur
CloudFormation laC.](#)

[Refactorisation de piles](#)

La refactorisation des piles
simplifie la réorganisation des
ressources de vos CloudForm
ation piles tout en préservant
les propriétés et les données
des ressources existante
s. Pour plus d'informations,
consultez [Refactorisation de
piles.](#)

6 février 2025

[Résolution des problèmes liés aux déploiements de piles avec Amazon Q Developer](#)

Vous pouvez désormais
utiliser Amazon Q Developer
pour résoudre les erreurs
courantes lors du déploieme
nt de CloudFormation stacks.
Pour plus d'informations,
consultez [Résoudre les
problèmes liés aux déploieme
nts de CloudFormation stack
infructueux avec Amazon Q
Developer.](#)

22 novembre 2024

[Graphique chronologique de déploiement d'une pile](#)

Vous pouvez désormais voir une représentation visuelle du déploiement de votre pile. Le graphique chronologique de déploiement affiche l'état du déploiement de la pile, l'état de déploiement de chaque ressource ainsi que les horaires auxquels ces états ont changé. Pour plus d'informations, voir [Afficher la chronologie d'un déploiement de CloudFormation stack](#).

11 novembre 2024

[Visualisation des ressources analysées et des modèles générés](#)

Vous pouvez désormais simplifier vos processus avec le générateur d'infrastructure en tant que code (IaC) en visualisant les détails récapitulatifs de l'analyse et en prévisualisant les modèles générés avant de déployer votre pile d'infrastructure. Pour plus d'informations, voir [Afficher le résumé de l'analyse dans la CloudFormation console](#) et [Créer une CloudFormation pile à partir de ressources numérisées](#).

22 août 2024

[EventBridge Intégration d'Amazon à AWS CloudFormation Git Sync](#)

AWS CloudFormation Git sync publie désormais les modifications du statut de synchronisation sous forme d'événements sur Amazon EventBridge. Pour plus d'informations, consultez [Détails de l'événement de changement d'état de synchronisation du référentiel](#) et [Détails de l'événement de changement d'état de synchronisation d'une ressource](#).

29 juillet 2024

[Suppression forcée des piles bloquées](#)

Deux nouvelles options de suppression forcée sont disponibles pour les opérations de suppression de piles qui restent bloquées. Vous pouvez désormais choisir de forcer la suppression de la pile tout en retenant la ressource, ou de forcer la suppression complète de la pile. Pour plus d'informations, consultez [Supprimer une pile de la CloudFormation console](#).

22 mai 2024

[AWS CloudTrail causes profondes de l'opération Event Stack](#)

CloudFormation améliore l'expérience de dépannage pour les opérations de stack grâce à une nouvelle intégration de AWS CloudTrail liens profonds. Cette fonctionnalité lie directement les événements liés aux opérations de pile dans la CloudFormation console aux CloudTrail événements pertinents. Pour plus d'informations, consultez [Détermination de la cause d'un échec de pile](#).

15 mai 2024

[Jeux de modifications au niveau des propriétés](#)

Les ensembles de modifications au niveau des propriétés vous permettent de prévisualiser les modifications que CloudFormation les déploiements apporteront aux valeurs de propriété des ressources. Pour plus d'informations, voir [Afficher un ensemble de modifications pour une CloudFormation pile](#).

12 avril 2024

[CloudFormation présente l'EVENT_CONFIGURATION_COMPLETED événement](#)

Vous pouvez désormais utiliser l'événement `EVENT_CONFIGURATION_COMPLETED` pour activer des flux de travail plus rapides impliquant la création de ressources. Pour plus d'informations, voir [Comprendre les événements de création de CloudFormation piles](#).

11 mars 2024

[Générez des AWS CloudFormation modèles et AWS CDK des applications à partir de AWS ressources existantes](#)

Vous pouvez désormais générer un modèle à l'aide de ressources fournies dans votre compte qui ne sont pas déjà gérées par CloudFormation. Pour plus d'informations, consultez [Génération de modèles à partir de ressources existantes avec le générateur laC](#).

2 février 2024

[StackSets mode simultané](#)

Le Mode simultané renvoie à un paramètre `StackSetOperationPreferences` qui vous permet de choisir le comportement du niveau de simultanéité lors des opérations d'ensemble de piles. Pour plus d'informations, voir [Choisir le mode de simultanéité pour CloudFormation StackSets](#).

9 novembre 2023

[Informations détaillées sur StackSet la dérive](#)

Ce qui suit vous APIs permet de voir quelles instances de pile ont dérivé du StackSet modèle et quelles ressources ont dérivé.

24 juillet 2023

[ListStackInstanceResourceDrifts](#)

Renvoie des informations sur les écarts pour les ressources d'une instance de pile.

[StackInstanceResourceDrifts Summary](#)

Structure contenant des informations récapitulatives sur les écarts de ressources pour une instance de pile.

[CloudFormation StackSets](#)
[APIs pour contrôler les](#)
[AWS Organizations accès](#)
[sécurisés](#)

CloudFormation StackSets fournit aux clients les informations suivantes API pour gérer l'accès de AWS Organizations confiance :

5 juin 2023

[ActivateOrganizationsAccess](#)

Activez un accès sécurisé avec AWS Organizations. Lorsque l'accès sécurisé entre StackSets et Organizations est activé, le compte de gestion dispose des autorisations nécessaires pour créer et gérer StackSets pour votre organisation.

[DeactivateOrganizationsAccess](#)

Désactive l'accès sécurisé avec AWS Organizations. Si l'accès sécurisé est désactivé, le compte de gestion n'est pas autorisé à créer et à gérer des services gérés StackSets pour votre organisation.

[DescribeOrganizationsAccess](#)

Récupération des informations sur le statut `OrganizationAccess` du compte. Cette API peut être appelée soit par le compte de gestion soit par

l'administrateur délégué à l'aide du paramètre `CallAs`. Cette API peut également être appelée sans le paramètre `CallAs` par le compte de gestion.

[API DescribeStackSet](#)

L'API `DescribeStackSet` ajoute un nouveau paramètre à la liste des régions dans lesquelles un ensemble de piles donné est déployé. Pour de plus amples informations, veuillez consulter [DescribeStackSet](#).

1er février 2023

[Gérer des StackSets événements avec Amazon CloudFormation et Amazon EventBridge](#)

CloudFormation StackSets notifications d'événements de lancement via Amazon EventBridge. Vous pouvez déclencher des actions pilotées par des événements après avoir créé, mis à jour ou supprimé vos ensembles de CloudFormation piles. Pour plus d'informations, consultez les [sections Surveillance CloudFormation et synchronisation des événements Git avec EventBridge](#).

16 novembre 2022

[Améliorations des informations relatives aux instances de pile pour les opérations sur les ensembles de piles](#)

CloudFormation StackSets fournit des informations plus détaillées sur les instances de stack pour les opérations de stack set :

4 novembre 2022

[DescribeStackSetOperation](#)

Vous pouvez désormais utiliser `DescribeStackSetOperation` pour fournir le nombre d'instances de pile ayant échoué pour les opérations sur les ensembles de piles lors du déploiement.

[ListStackInstances](#)

Vous pouvez désormais utiliser l'option de filtrage `LastOperationID` pour répertorier les instances de pile pour les opérations sur les ensembles de piles.

[Gérer des événements avec Amazon CloudFormation et Amazon EventBridge](#)

Recevez des notifications lorsque des CloudFormation événements spécifiques se produisent. Pour plus d'informations, consultez les [sections Surveillance CloudFormation et synchronisation des événements Git avec EventBridge](#).

20 juillet 2022

Niveau du compte

CloudFormation annonce la disponibilité générale du type de filtre de compte, une fonctionnalité qui permet aux clients de limiter les cibles de déploiement à des comptes individuels ou d'inclure des comptes supplémentaires avec provided OUs. Pour plus d'informations, consultez la section [Objectifs au niveau du compte pour les services gérés. StackSets](#)

7 juillet 2022

CloudFormation registre

CloudFormation annonce la disponibilité générale de Hooks, une fonctionnalité qui permet aux clients d'invoquer une logique personnalisée pour automatiser des actions ou inspecter les configurations des ressources avant une opération de création, de mise à jour ou de suppression de pile. Pour plus d'informations, consultez le [Guide de l'utilisateur des hooks CloudFormation](#).

10 février 2022

[Options d'échec de pile](#)

Vous pouvez développer vos applications de manière itérative en cas d'échec d'allocation en partant du point de défaillance sans annuler les ressources allouées avec succès. En spécifiant les options d'échec de pile, vous pouvez dépanner les ressources dans un état `CREATE_FAILED` ou `UPDATE_FAILED`.

30 août 2021

Vous pouvez allouer des options d'échec pour tous les déploiements de pile et les opérations d'ensembles de modifications. Pour plus d'informations, consultez [Choisir comment gérer les échecs lors du provisionnement des ressources](#).

[Importer des piles dans un ensemble de piles](#)

Vous pouvez désormais importer des piles existantes dans des ensembles de piles nouveaux ou existants. Pour plus d'informations, consultez la section [Importation de piles dans CloudFormation StackSets](#).

28 juillet 2021

Augmentation du quota	Vous pouvez maintenant déclarer un maximum par défaut de piles 2000 dans votre Compte AWS. Pour plus d'informations, voir Comprendre les CloudFormation quotas.	15 juillet 2021
Publier des extensions tierces publiques	Vous pouvez désormais utiliser les extensions publiques fournies par des éditeurs tiers, comme vous le feriez avec les extensions de AWS. Pour plus d'informations, voir Utiliser des extensions publiques tierces depuis le CloudFormation registre.	21 juin 2021
Macros de référence dans les modèles d'ensemble de piles	StackSets prend désormais en charge la création ou la mise à jour d'ensembles de piles avec des autorisations autogérées à partir de modèles faisant référence à des macros. Pour plus d'informations sur les macros, voir Effectuer un traitement personnalisé sur des CloudFormation modèles à l'aide de modèles de macros.	14 avril 2021

[Utilisation de la valeur la plus récente d'un paramètre Systems Manager dans une référence dynamique](#)

Vous pouvez désormais CloudFormation utiliser la dernière version d'un paramètre de Systems Manager chaque fois que vous créez ou mettez à jour une pile. Vous n'êtes plus tenu de spécifier une version spécifique. Pour plus de détails, consultez [Obtenir une valeur en texte clair à partir du magasin de paramètres Systems Manager](#).

13 avril 2021

[Les modules prennent en charge l'utilisation de délimiteurs de période dans les noms de ressources](#)

Vous pouvez désormais utiliser un point comme délimiteur pour spécifier le nom logique complet d'une ressource contenue dans un module. Pour plus d'informations, consultez la section [Ressources du module de référence dans les CloudFormation modèles](#).

08 avril 2021

[CloudFormation StackSets prend désormais en charge le déploiement par région parallèle](#)

Vous pouvez désormais choisir de déployer StackSets dans les régions de manière séquentielle ou en parallèle. Pour plus d'informations, consultez la section [Options d'opération d'ensemble de piles](#).

6 avril 2021

[CloudFormation StackSets prend désormais en charge l'administrateur délégué avec AWS Organizations](#)

Outre les compte de gestion de l'organisation, les comptes d'administrateur délégué peuvent créer et gérer des ensembles de piles avec des autorisations gérées par le service pour leur organisation. Pour plus d'informations, voir [Enregistrer un compte de membre administrateur délégué](#) et [Créer CloudFormation StackSets avec des autorisations gérées par le service](#).

18 février 2021

[CloudFormation StackSets Disponibilité dans les Régions](#)

CloudFormation StackSets est désormais disponible dans la région Asie-Pacifique (Osaka). Pour plus d'informations, consultez [la section Gestion des piles entre les comptes et les régions avec StackSets](#).

10 février 2021

Modules

Les modules vous permettent d'empaqueter des configurations de ressources pour les inclure dans les modèles de pile, de manière transparente, gérable et reproductible. Les modules peuvent encapsuler les configurations de service courantes et les bonnes pratiques en tant que blocs de création modulaires et personnalisables à inclure dans vos modèles de pile. Pour plus d'informations, voir [Création de configurations de ressources réutilisables pouvant être incluses dans les modèles comportant des CloudFormation modules](#).

24 novembre 2020

Modifier les ensembles pour les piles imbriquées

Avec les ensembles de modifications pour les piles imbriquées, vous pouvez prévisualiser les modifications apportées à vos ressources d'application et d'infrastructure sur l'ensemble de la hiérarchie de pile imbriquée et procéder aux mises à jour lorsque vous avez confirmé que toutes les modifications sont celles que vous comptiez. Pour obtenir plus d'informations, consultez la section [Modifier les ensembles de piles imbriquées](#).

18 novembre 2020

Augmentation des quotas

Les AWS CloudFormation quotas suivants ont été mis à jour.

22 octobre 2020

- Vous pouvez maintenant déclarer un maximum de 200 mappages dans votre modèle AWS CloudFormation .
- Vous pouvez désormais déclarer un maximum d'attributs de 200 mappage pour chaque mappage de votre AWS CloudFormation modèle.
- Vous pouvez maintenant déclarer un maximum de sorties 200 dans votre modèle AWS CloudFormation .
- Vous pouvez maintenant déclarer un maximum de paramètres 200 dans votre modèle AWS CloudFormation .
- Vous pouvez maintenant déclarer un maximum de ressources 500 dans votre modèle AWS CloudFormation .
- Vous pouvez désormais passer un corps de modèle avec une taille maximale de 1 MB dans un objet Amazon S3.

[Détection de l'écart pour les ressources privées](#)

CloudFormation prend désormais en charge les opérations de détection de dérive sur une liste étendue de AWS ressources, ainsi que les ressources privées définies comme provisoires dans le CloudFormation registre. Pour plus d'informations, consultez [Prise en charge des types de ressources](#).

1er octobre 2020

[Autorisations mises à jour requises pour l'enregistrement des fournisseurs de ressources](#)

L'inscription d'un fournisseur de ressources dans votre compte nécessite désormais l'autorisation d'accéder au package de gestionnaire de schéma téléchargé dans un compartiment S3 pour ce fournisseur de ressources. Pour plus d'informations, consultez [Autorisations IAM pour l'enregistrement d'une extension privée tierce](#).

7 août 2020

[L'importation de ressources prend en charge les types de ressource privée allouable](#)

Les opérations d'importation prennent désormais en charge les types de ressource privée pouvant être provisionnés, c'est-à-dire, dont le type de mise en service est `FULLY_MUTABLE` ou `IMMUTABLE`. Pour de plus amples informations, veuillez consulter [Ressources prenant en charge les opérations d'importation](#).

3 juin 2020

[blue/green Déploiements ECS via CodeDeploy](#)

Vous pouvez désormais utiliser CloudFormation pour effectuer des blue/green déploiements ECS via CodeDeploy. Blue/green les déploiements sont une stratégie de déploiement sûre fournie AWS CodeDeploy pour minimiser les interruptions causées par le changement de version des applications. Pour plus d'informations, voir [Exécution de blue/green déploiements ECS CodeDeploy en utilisant CloudFormation](#).

19 mai 2020

[CloudFormation StackSets Disponibilité dans les Régions](#)

CloudFormation StackSets est désormais disponible dans la région AWS GovCloud (ouest des États-Unis).

18 mai 2020

[AWS CloudFormation StackSets s'intègre à AWS Organizations](#)

Vous pouvez désormais utiliser StackSets pour gérer de manière centralisée les déploiements vers tous les comptes de votre organisation ou vers des unités organisationnelles spécifiques (OUs) dans AWS Organizations. Vous pouvez activer les déploiements automatiques vers tous les nouveaux comptes ajoutés à votre organisation ou OUs. Les autorisations nécessaires au déploiement sur plusieurs comptes seront automatiquement gérées par StackSets. Pour plus d'informations, consultez [la section Gestion des piles entre les comptes et les régions avec StackSets](#).

11 février 2020

[Détection de dérive pour StackSets](#)

Vous pouvez désormais exécuter la détection de l'écart sur un ensemble de piles et toutes les instances de pile qu'il inclut. Pour plus d'informations, consultez la section [Exécution de la détection de dérive sur CloudFormation StackSets](#).

19 novembre 2019

[CloudFormation le registre est maintenant disponible](#)

Vous pouvez désormais utiliser la CloudFormation console pour consulter les ressources privées et publiques disponibles sur votre compte. Pour plus d'informations, voir [Afficher les extensions disponibles et activées dans le CloudFormation registre](#).

18 novembre 2019

[CloudFormation actions de l'API de registre](#)

Les actions d'API suivantes pour gérer les types dans le CloudFormation registre sont désormais disponibles.

18 novembre 2019

[DeregisterType](#)

Supprime un type ou une version de type d'une utilisation active dans le CloudFormation registre.

[DescribeType](#)

Renvoie des informations détaillées sur un type enregistré.

[DescribeTypeRegistration](#)

Renvoie des informations sur l'enregistrement d'un type, y compris son état actuel et ses identifiants de type et de version.

[ListTypeRegistrations](#)

Renvoie une liste d'identifiants de demande d'enregistrement pour le type spécifié.

[ListTypes](#)

Renvoie des informations récapitulatives sur les types enregistrés auprès de CloudFormation.

[ListTypeVersions](#)

Renvoie des informations récapitulatives sur les versions d'un type.

[RegisterType](#)

Enregistre un type dans le CloudFormation registre. L'enregistrement d'un type permet de l'utiliser dans les CloudFormation modèles de votre Compte AWS.

[SetTypeDefaultVersion](#)

Spécifiez la version par défaut d'un type. La version par défaut d'un type sera utilisée dans les CloudFormation opérations.

Pour plus d'informations sur le CloudFormation registre, voir [Gestion des extensions avec le CloudFormation registre](#)

[Importation de ressources ajoutée](#)

Si vous avez créé une AWS ressource en dehors de la CloudFormation gestion, vous pouvez intégrer cette ressource existante à CloudFormation la gestion à l'aide de ressource import. Pour plus d'informations, voir [Importer AWS des ressources dans une CloudFormation pile à l'aide d'une importation de ressources](#).

11 novembre 2019

[Augmentation des limites des ensembles de piles](#)

Vous pouvez désormais créer un maximum de 100 ensembles de piles dans votre compte administrateur, créer un maximum de 2 000 instances de piles par ensemble de piles et exécuter un maximum de 3 500 opérations d'instance de piles dans chaque région en même temps, par compte administrateur. Pour plus d'informations, voir [Comprendre les CloudFormation quotas](#).

2 août 2019

Limite de ressources lors d'opérations de stack simultanées

CloudFormation impose désormais une limite de compte pour le nombre de ressources dans le cadre d'opérations de pile simultanées. Cette limite est déterminée par région. Pour plus d'informations, voir [Comprendre les CloudFormation quotas](#).

30 avril 2019

Limite liée aux opérations d'instance de pile

En StackSets effect, vous pouvez désormais exécuter simultanément un maximum de 1 500 opérations d'instance de stack dans une région donnée, par compte administrateur. Pour plus d'informations, voir [Comprendre les CloudFormation quotas](#).

13 décembre 2018

La CAPABILITY_AUTO_EXPAND fonctionnalité est désormais disponible

Vous pouvez utiliser la fonctionnalité CAPABILITY_AUTO_EXPAND pour créer ou de mettre à jour une pile directement à partir d'un modèle de pile qui contient des macros, sans devoir d'abord examiner les modifications résultantes dans un jeu de modifications. Pour plus d'informations, consultez la section [CreateStack](#) (français non garanti) ou la section [UpdateStack](#) (français non garanti) dans la Référence d'API AWS CloudFormation .

7 décembre 2018

[Ajout de la détection de l'écart de pile](#)

13 novembre 2018

Vous pouvez désormais détecter si la configuration réelle d'une pile s'est éloignée de la configuration de modèle attendue, telle que définie dans CloudFormation ce document. La détection de dérive peut être effectuée sur l'ensemble d'une pile ou sur des ressources individuelles. Pour plus d'informations, consultez [Détection des modifications de configuration non gérées dans les piles et les ressources à l'aide de la détection de dérive.](#)

[La référence dynamique secretsmanager est désormais disponible](#)

Vous pouvez maintenant utiliser la référence dynamique `secretsmanager` pour extraire des secrets complets ou des valeurs de secrets stockés dans AWS Secrets Manager. Les secrets peuvent être des informations d'identification de base de données, des mots de passe, des clés d'API tierces et même un texte arbitraire. L'utilisation de la référence `secretsmanager` dynamique garantit que ni Secrets Manager ni CloudFormation enregistre ou ne conserve aucune valeur secrète résolue. Pour plus d'informations, consultez [Obtenir un secret ou une valeur secrète depuis Secrets Manager](#).

9 novembre 2018

[Macros désormais disponibles](#)

Vous pouvez désormais utiliser des macros pour effectuer un traitement personnalisé sur les modèles, qu'il s'agisse d'actions simples telles que find-and-replace des opérations ou de transformations étendues de modèles entiers. Pour plus d'informations, voir [Effectuer un traitement personnalisé sur des CloudFormation modèles à l'aide de macros de modèles](#).

6 septembre 2018

[CloudFormation prend désormais en charge les points de terminaison VPC alimentés par PrivateLink](#)

Vous pouvez utiliser le point de terminaison d'un VPC pour créer une connexion privée entre votre VPC et CloudFormation sans avoir besoin d'un accès Internet, via une instance NAT, une connexion VPN ou Direct Connect. Pour plus d'informations, consultez [Accès à CloudFormation l'aide d'un point de terminaison d'interface \(AWS PrivateLink\)](#).

22 août 2018

[Les références dynamiques prennent en charge les chaînes sécurisées](#)

Vous pouvez désormais utiliser de nouvelles références dynamiques pour spécifier, dans vos modèles de pile, des valeurs stockées et gérées dans d'autres services, y compris les paramètres de type SecureString du magasin de paramètres Systems Manager. Pour plus d'informations, consultez [Obtenir une valeur sécurisée à partir du magasin de paramètres Systems Manager](#).

16 août 2018

[Les ensembles de piles prennent désormais en charge les rôles d'exécution](#)

Vous pouvez désormais utiliser des rôles d'exécution personnalisés dans les comptes cibles afin de contrôler les ressources de pile que les utilisateurs ou groupes peuvent inclure dans leurs ensembles de piles. Pour plus d'informations, consultez [Configuration des options d'autorisations avancées pour les opérations sur les ensembles de piles](#).

30 mai 2018

[Mises à jour sélectives des instances de piles](#)

Vous pouvez désormais utiliser les paramètres Accounts et Regions optionnels permettant de spécifier les comptes et les régions dans lesquels mettre à jour les instances de pile lors d'une opération de mise à jour d'un ensemble de piles. Pour plus d'informations, consultez [UpdateStackSet](#) dans la Référence d'API CloudFormation .

30 mai 2018

[CloudFormation crée désormais des compartiments S3 avec le chiffrement activé](#)

Pour les compartiments Amazon S3 qui CloudFormation créent pour stocker des modèles de stack téléchargés, le chiffrement côté serveur est désormais activé par défaut, chiffrant ainsi tous les objets stockés dans ces compartiments. Pour plus d'informations, consultez la section [Créer une pile depuis la CloudFormation console](#).

24 mai 2018

[Points de terminaison FIPS ajoutés](#)

CloudFormation propose désormais de nouveaux points de terminaison qui utilisent des modules cryptographiques validés par la norme FIPS 140-2 dans les régions publiques américaines suivantes : US-East-1, US-East-2, US-West-1 et US-West-2. Voir les [CloudFormation points de terminaison et les quotas](#) dans le Référence générale d'Amazon Web Services pour le nouveau point de terminaison conforme à la norme FIPS. URLs

17 mai 2018

Pour les mises à jour du guide de l'utilisateur de CloudFormation Hooks, consultez [l'historique des documents du guide de l'utilisateur de CloudFormation Hooks dans le guide](#) de l'utilisateur de CloudFormation Hooks.

Mises à jour archivées

Le tableau suivant décrit les modifications importantes apportées à chaque version du guide de AWS CloudFormation l'utilisateur avant mai 2018.

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	22 juillet 2019	Utilisez la propriété <code>encryptionOptions</code> pour spécifier une Clé détenue par AWS ou une clé KMS gérée par le client pour les agents Amazon MQ.	2010-05-15
Convention de dénominat	10 avril 2018	CloudFormation les piles créées à l'aide d'ensembles de piles suivent désormais une nouvelle convention de	2010-05-15

Modifier	Date de parution	Description	Version de l'API
ion des ensembles de piles		dénomination, dans laquelle le nom de la pile contient le nom de l'ensemble de piles.	
Nouvelles ressources	10 avril 2018	AWS::AppSync::ApiKey Utilisez cette <code>AWS::AppSync::ApiKey</code> ressource pour créer une clé unique que vous pouvez distribuer aux clients qui exécutent des opérations GraphQL avec. AWS AppSync AWS::AppSync::DataSource Utilisez la <code>AWS::AppSync::DataSource</code> ressource pour créer des sources de données pour les résolveurs dans AWS AppSync. AWS::AppSync::GraphQLApi Utilisez la <code>AWS::AppSync::GraphQLApi</code> ressource pour créer une nouvelle API AWS AppSync GraphQL. AWS::AppSync::GraphQLSchema Utilisez la <code>AWS::AppSync::GraphQLSchema</code> ressource pour créer le modèle de données pour votre API AWS AppSync GraphQL. AWS::AppSync::Resolver Utilisez la ressource <code>AWS::AppSync::Resolver</code> pour définir le résolveur GraphQL logique que vous allez attacher aux champs dans un schéma.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	10 avril 2018	AWS::Config::ConfigurationAggregator Utilisez le type de <code>OrganizationAggregationSource</code> propriété pour spécifier les régions de AWS Config données à agréger dans un agrégateur de AWS Config configuration et le rôle IAM à utiliser pour récupérer les AWS Organizations détails.	2010-05-15
Nouvelles ressources	4 avril 2018	AWS::Config::AggregationAuthorization Utilisez la <code>AWS::Config::AggregationAuthorization</code> ressource pour autoriser un compte agrégateur à collecter vos AWS Config données. AWS::Config::ConfigurationAggregator La ressource <code>AWS::Config::ConfigurationAggregator</code> permet de créer un agrégateur de configuration pour AWS Config.	2010-05-15
Les ensembles de piles prennent désormais en charge les rôles d'administrateur personnalisés	29 mars 2018	Utilisez des rôles d'administrateur personnalisés pour contrôler les utilisateurs ou groupes qui peuvent gérer des ensembles de piles spécifiques au sein d'un même compte d'administrateur. Pour plus d'informations, consultez Configuration des options d'autorisations avancées pour les opérations sur les ensembles de piles.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	29 mars 2018	AWS::EC2::LaunchTemplate La ressource AWS::EC2::LaunchTemplate permet de créer un modèle de lancement pour une instance Amazon EC2.	2010-05-15
Ressources mises à jour	29 mars 2018	AWS::AutoScaling::AutoScalingGroup Utilisez la propriété LaunchTemplate pour spécifier le modèle de lancement à utiliser pour lancer des instances. AWS::EC2::SpotFleet Dans le type de propriété SpotFleetRequestConfigurationData , utilisez la propriété LaunchTemplateConfigs pour décrire un modèle de lancement et les remplacements.	2010-05-15
Nouvelle fonction intrinsèque Fn::Cidr	6 mars 2018	Renvoie le bloc d'adresses Cidr spécifié. Pour de plus amples informations, veuillez consulter Fn::Cidr .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	6 mars 2018	AWS::ApiGateway::VpcLink Utilisez la ressource <code>AWS::ApiGateway::VpcLink</code> pour spécifier un lien VPC API Gateway pour qu'une <code>AWS::ApiGateway::RestApi</code> accède aux ressources présentes dans un Amazon Virtual Private Cloud (VPC). AWS::GuardDuty::Master Utilisez la <code>AWS::GuardDuty::Master</code> ressource pour créer un compte GuardDuty principal. AWS::GuardDuty::Member Utilisez la <code>AWS::GuardDuty::Member</code> ressource pour créer un compte GuardDuty membre. AWS::SES::ConfigurationSet Utilisez la ressource <code>AWS::SES::ConfigurationSet</code> pour créer des groupes de règles que vous pouvez appliquer aux e-mails que vous envoyez. AWS::SES::ConfigurationSetEventDestination Utilisez la ressource <code>AWS::SES::ConfigurationSetEventDestination</code> pour spécifier une destination d'événement de jeu de configurations. AWS::SES::ReceiptFilter Utilisez la ressource <code>AWS::SES::ReceiptFilter</code> pour spécifier si les messages provenant d'une adresse IP ou d'une plage d'adresses IP doivent être acceptés ou rejetés.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		<u>AWS::SES::ReceiptRule</u> La ressource <code>AWS::SES::ReceiptRule</code> permet de spécifier les actions que doit effectuer Amazon SES à réception d'e-mails au nom d'un ou plusieurs domaines ou adresses e-mail dont vous êtes propriétaire. <u>AWS::SES::ReceiptRuleSet</u> La ressource <code>AWS::SES::ReceiptRuleSet</code> permet de spécifier un ensemble de règles vide pour Amazon SES. <u>AWS::SES::Template</u> Utilisez la ressource <code>AWS::SES::Template</code> pour spécifier le contenu de l'e-mail, composé d'une ligne d'objet, d'une partie HTML et d'une partie du texte uniquement.	

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	6 mars 2018	AWS::AutoScaling::AutoScalingGroup Utilisez la propriété <code>AutoScalingGroupName</code> pour spécifier le nom du groupe Auto Scaling. AWS::ApiGateway::RestApi Utilisez la propriété <code>ApiKeySourceType</code> pour spécifier la source de la clé API pour les demandes de relevé en fonction d'un plan d'utilisation. Utilisez la propriété <code>MinimumCompressionSize</code> pour spécifier un nombre entier de type nullable pour activer ou désactiver la compression sur une API. AWS::ApplicationAutoScaling::ScalingPolicy Dans le type de propriété TargetTrackingScalingPolicyConfiguration, utilisez la propriété <code>DisableScaleIn</code> pour spécifier si la mise à l'échelle par la stratégie de dimensionnement Suivi de la cible est désactivée. AWS::EC2::SpotFleet Dans le type de LaunchSpecifications propriété, utilisez la <code>TagSpecifications</code> propriété pour spécifier les balises à appliquer lors de SpotFleet la création. AWS::Elasticsearch::Domain Utilisez l'attribut <code>Arn</code> pour que <code>Fn::GetAtt</code> renvoie l'Amazon Resource Name (ARN) du domaine. L'utilisation de l'attribut <code>DomainArn</code> de <code>Fn::GetAtt</code> est devenue obsolète.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::RDS::DBCluster Utilisez la propriété <code>DBClusterIdentifier</code> pour spécifier l'identifiant du cluster de bases de données. AWS::RDS::DBCluster Utilisez la propriété <code>DBClusterIdentifier</code> pour spécifier l'identifiant du cluster de bases de données. AWS::Redshift::Cluster Utilisez la propriété <code>ClusterIdentifier</code> pour spécifier l'identifiant unique du cluster. AWS::Route53::HealthCheck Dans le type de HealthCheckConfig propriété, utilisez la <code>Regions</code> propriété pour spécifier les régions à partir desquelles vous souhaitez que les vérificateurs de santé Route 53 vérifient le point de terminaison spécifié. AWS::SSM::Document La propriété <code>Tags</code> permet de spécifier les balises de ressource CloudFormation à appliquer au document.	
Ressource mise à jour	19 février 201	AWS::CodeBuild::Project Utilisez la propriété <code>Triggers</code> pour configurer un webhook afin que le projet commence automatiquement à régénérer le code source chaque fois qu'une modification de code est transmise au référentiel. Ceci n'est disponible que pour les GitHub projets en CloudFormation. Il n'est pas disponible pour les projets GitHub d'entreprise.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	8 février 2018	AWS::DynamoDB::Table Utilisez la propriété <code>SSESpecification</code> pour spécifier les paramètres visant à activer le chiffrement côté serveur.	2010-05-15
Ressource mise à jour	5 février 2018	AWS::CodeBuild::Project Dans le type de propriété Source CodeBuild Project Source : • Utilisez la propriété <code>GitCloneDepth</code> pour spécifier la profondeur de l'historique à télécharger. • Utilisez cette <code>InsecureSsl</code> propriété pour indiquer si les avertissements SSL doivent être ignorés lors de la connexion au référentiel de votre projet GitHub Enterprise.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	23 janvier 201	AWS::AutoScaling::LifecycleHook Utilisez la propriété <code>LifecycleHookName</code> pour spécifier le nom du hook de cycle de vie. AWS::DynamoDB::Table La propriété <code>AttributeDefinitions</code> doit être remplacée lors de la mise à jour. AWS::EC2::Instance Utilisez la propriété <code>CreditSpecification</code> pour spécifier l'option relative aux crédits pour l'utilisation de l'UC d'une instance T2. Utilisez cette <code>ElasticGpuSpecifications</code> propriété pour spécifier les ressources GPU élastiques GPUs que vous pouvez associer à votre instance afin d'accélérer les performances graphiques de vos applications. AWS::EC2::VPC La propriété <code>InstanceTenancy</code> ne nécessite désormais aucune interruption lors de la mise à jour de "dedicated" vers "default" . AWS::ECS::Service La propriété <code>HealthCheckGracePeriodSeconds</code> permet de spécifier le laps de temps en secondes durant lequel le planificateur de service Amazon ECS ignore les vérifications de l'état des cibles Elastic Load Balancing non saines après le démarrage préalable d'une tâche.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::IoT::TopicRule Dans le type de propriété DynamoDBAction, les propriétés <code>RangeKeyField</code> et <code>RangeKeyValue</code> ne sont plus obligatoires. AWS::KinesisAnalytics::ApplicationOutput Dans le type de propriété ApplicationOutput, la propriété <code>LambdaOutput</code> permet d'identifier une fonction Lambda comme destination lors de la configuration de la sortie de l'application. AWS::Kinesis::Stream Utilisez cette <code>StreamEncryption</code> propriété pour activer ou mettre à jour le chiffrement côté serveur à l'aide d'un AWS KMS key pour un flux spécifique. AWS::Lambda::Function Utilisez la propriété <code>ReservedConcurrentExecutions</code> pour spécifier le nombre maximum d'exécutions simultanées à réserver pour la fonction. AWS::RDS::DBSubnetGroup Utilisez la propriété <code>DBSubnetGroupName</code> pour spécifier le nom du groupe de sous-réseaux DB. AWS::S3::Bucket Utilisez cette <code>BucketEncryption</code> propriété pour spécifier le chiffrement par défaut d'un compartiment à l'aide du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) ou un bucket (SSE-KMS). AWS KMS keys	

Modifier	Date de parution	Description	Version de l'API
		Dans le type de propriété ReplicationRule, la propriété <code>SourceSelectionCriteria</code> permet de spécifier des filtres supplémentaires dans l'identification des objets source que vous souhaitez répliquer. Dans le type de propriété ReplicationDestination : - Utilisez la propriété <code>AccessControlTranslation</code> pour spécifier la propriété de la réplique du AWS compte propriétaire du compartiment de destination. - Utilisez la propriété <code>Account</code> pour spécifier l'ID du compte du propriétaire du compartiment de destination. - Utilisez la propriété <code>EncryptionConfiguration</code> pour spécifier des informations de chiffrement pour un compartiment qui est une destination pour les objets répliqués. AWS::SSM::Association Utilisez la propriété <code>AssociationName</code> pour spécifier le nom de l'association entre un document SSM et les instances EC2 qui contiennent un agent de configuration pour traiter le document.	
Des déclencheurs de restauration ont été ajoutés à la CloudFormation console.	15 janvier 201	Les déclencheurs d'annulation vous permettent de CloudFormation surveiller l'état de votre application lors de la création et de la mise à jour de la pile, et d'annuler cette opération si l'application dépasse le seuil de l'une des alarmes que vous avez spécifiées. Pour plus d'informations, consultez Surveillance et restauration des opérations sur la pile .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	12 janvier 201	AWS::SSM::Parameter Utilisez la propriété <code>AllowedPattern</code> pour spécifier une expression régulière utilisée pour valider la valeur du paramètre.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	5 décembre 2017	AWS::Inspector::AssessmentTarget Utilisez la ressource <code>AWS::Inspector::AssessmentTarget</code> pour créer un objectif d'évaluation Amazon Inspector. AWS::Inspector::AssessmentTemplate Utilisez la ressource <code>AWS::Inspector::AssessmentTemplate</code> pour créer un modèle d'évaluation Amazon Inspector. AWS::Inspector::ResourceGroup Utilisez la ressource <code>AWS::Inspector::ResourceGroup</code> pour créer un groupe de ressources Amazon Inspector définissant les balises qui identifient les ressources AWS constituant un objectif d'évaluation Amazon Inspector. AWS::ServiceDiscovery::Instance La ressource <code>AWS::ServiceDiscovery::Instance</code> permet de spécifier les informations concernant une instance créée par Amazon Route 53. AWS::ServiceDiscovery::PrivateDnsNamespace La ressource <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> permet de spécifier des informations concernant un espace de noms privé pour Amazon Route 53. AWS::ServiceDiscovery::PublicDnsNamespace La ressource <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> permet de spécifier des	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		informations concernant un espace de noms public pour Amazon Route 53. AWS::ServiceDiscovery::Service La ressource <code>AWS::ServiceDiscovery::Service</code> permet de définir un modèle pour cinq enregistrements au maximum et une vérification de l'état facultative que créera Amazon Route 53 lorsque vous inscrirez une instance.	
Ressource mise à jour	5 décembre 2017	AWS::KinesisAnalytics::Application Dans le type de propriété Input, la propriété <code>InputProcessingConfiguration</code> permet de transformer les enregistrements à mesure qu'ils sont reçus en provenance du flux.	2010-05-15
Ressource mise à jour	1er décembre 2017	AWS::CodeBuild::Project Utilisez la propriété <code>BadgeEnabled</code> afin de générer une URL publiquement accessible pour un badge de compilation de projet. Utilisez la propriété <code>Cache</code> afin de configurer les paramètres du cache pour les dépendances de compilation. Utilisez la <code>VpcConfig</code> propriété pour activer CodeBuild l'accès aux ressources d'un Amazon VPC. Dans le type de propriété EnvironmentVariable, la propriété <code>Type</code> permet de spécifier le type de variable d'environnement.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	30 novembre	AWS::Cloud9::EnvironmentEC2 Utilisez la ressource <code>AWS::Cloud9::EnvironmentEC2</code> pour créer un environnement de développement Amazon EC2 dans AWS Cloud9.	2010-05-15
Ressources mises à jour	29 novembre	AWS::ECS::TaskDefinition Utilisez la propriété <code>Cpu</code> pour spécifier le nombre d'unités UC nécessaires pour la tâche. Utilisez la propriété <code>ExecutionRoleArn</code> pour spécifier l'ARN du rôle d'exécution. Utilisez la propriété <code>Memory</code> afin de préciser la quantité de mémoire (en MiO) nécessaire pour la tâche. Utilisez la propriété <code>RequiresCompatibilities</code> afin de préciser le type de lancement nécessaire pour la tâche. AWS::ECS::Service Utilisez la propriété <code>LaunchType</code> pour spécifier le type de lancement sur lequel vous voulez exécuter votre service. Utilisez la propriété <code>NetworkConfiguration</code> pour préciser la configuration de réseau du service. Utilisez la propriété <code>PlatformVersion</code> pour préciser la version de plate-forme sur laquelle exécuter votre service.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	28 novembre	AWS::GuardDuty::Detector Utilisez cette <code>AWS::GuardDuty::Detector</code> ressource pour créer un GuardDuty détecteur Amazon unique. AWS::GuardDuty::IPSet Utilisez la <code>AWS::GuardDuty::IPSet</code> ressource pour créer un ensemble d' GuardDutyadresses IP Amazon. AWS::GuardDuty::ThreatIntelSet Utilisez la <code>AWS::GuardDuty::ThreatIntelSet</code> ressource pour créer un ThreatIntelSet.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	28 novembre	AWS::CodeDeploy::Application Utilisez la <code>ComputePlatform</code> propriété pour spécifier une plate-forme de AWS Lambda calcul sur laquelle CodeDeploy déployer une application. AWS::CodeDeploy::DeploymentGroup Dans le type de propriété DeploymentStyle, la propriété <code>DeploymentType</code> permet de spécifier un déploiement bleu/vert sur une plateforme de calcul Lambda. AWS::EC2::SpotFleet Dans le type de propriété SpotFleetRequestConfigurationData, la propriété <code>SpotPrice</code> est maintenant facultative. AWS::Lambda::Alias Utilisez cette <code>RoutingConfig</code> propriété pour spécifier deux versions différentes d'une AWS Lambda fonction, ce qui vous permet de dicter le pourcentage de trafic qui invoquera chaque version.	2010-05-15
Nouvelle politique de mise à jour <code>CodeDeployLambdaAliasUpdate</code>	28 novembre	Utilisez la politique de <code>CodeDeployLambdaAliasUpdate</code> mise à jour pour effectuer un CodeDeploy déploiement lorsque la version d'une <code>AWS::Lambda::Alias</code> ressource change. Pour plus d'informations, consultez Attribut UpdatePolicy.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouveaux types de paramètres SSM	21 novembre	Les types de paramètre SSM permettent d'utiliser les paramètres existants de Systems Manager Parameter Store. Remarque : CloudFormation ne prend pas actuellement en charge le type <code>SecureString</code> . Pour plus d'informations, consultez Types de paramètres SSM .	2010-05-15
Nouveau champ <code>ResolvedValue</code> pour le type de donnée <code>Parameter</code>	21 novembre	Le champ <code>ResolvedValue</code> renvoie la valeur utilisée dans la définition de pile pour un paramètre SSM. Pour plus d'informations, consultez le type de données Parameter (français non garanti) dans le document Référence d'API CloudFormation .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	le 20 novembre 2017	AWS::ApiGateway::ApiKey Utilisez la propriété <code>CustomerId</code> pour spécifier un identifiant de client AWS Marketplace. Utilisez la propriété <code>GenerateDistinctId</code> pour spécifier si l'identifiant de clé est distinct de la valeur de la clé API créée. AWS::ApiGateway::Authorizer Utilisez la propriété <code>AuthType</code> pour préciser un champ défini par le client qui est utilisé dans les importations et exportations Swagger sans impact fonctionnel. AWS::ApiGateway::DomainName La propriété <code>EndpointConfiguration</code> permet de spécifier les types de point de terminaison d'un nom de domaine API Gateway. Utilisez la propriété <code>RegionalCertificateArn</code> afin de référencer un certificat pour une utilisation par le point de terminaison régional d'un nom de domaine. AWS::ApiGateway::Method Dans les types de propriété Integration et IntegrationResponse, utilisez la propriété <code>ContentHandling</code> pour spécifier la façon dont sont traitées les conversions de type de contenu de données utiles de la requête.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::ApiGateway::RestApi La propriété <code>EndpointConfiguration</code> permet de spécifier les types de point de terminaison d'une API REST API Gateway. AWS::ApplicationAutoScaling::ScalableTarget La propriété <code>ScheduledActions</code> permet de spécifier les actions planifiées pour une cible scalable Application Auto Scaling. AWS::ECR::Repository La propriété <code>LifecyclePolicy</code> permet de spécifier une politique de cycle de vie pour un référentiel Amazon ECR. AWS::ECS::TaskDefinition Dans le type de propriété ContainerDefinition, la propriété <code>LinuxParameters</code> permet de spécifier des options propres à Linux pour un conteneur Amazon ECS. AWS::ElastiCache::ReplicationGroup Utilisez la propriété <code>AtRestEncryptionEnabled</code> pour activer le chiffrement au repos. Utilisez la propriété <code>AuthToken</code> afin de préciser un mot de passe pour accéder à un serveur protégé par mot de passe. Utilisez la propriété <code>TransitEncryptionEnabled</code> pour activer le chiffrement en transit.	

Modifier	Date de parution	Description	Version de l'API
		AWS::ElasticLoadBalancingV2::TargetGroup Utilisez l'attribut <code>TargetGroupName</code> avec la fonction <code>Fn::GetAtt</code> pour obtenir le nom d'un groupe cible Elastic Load Balancing. AWS::Elasticsearch::Domain Utilisez la <code>VPCOptions</code> propriété pour spécifier une configuration VPC pour le domaine de OpenSearch service. AWS::EMR::Cluster La propriété <code>EbsRootVolumeSize</code> permet de spécifier la taille du volume racine EBS pour un cluster Amazon EMR. AWS::RDS::DBInstance Utilisez les propriétés <code>SourceRegion</code> et <code>KmsKeyId</code> pour créer un réplica en lecture entre régions à partir d'une instance de base de données source. AWS::Route53::HostedZone Utilisez la propriété <code>QueryLoggingConfig</code> afin de préciser une configuration pour l'enregistrement de requêtes DNS.	
Nouveau champ <code>NoEcho</code> pour les objets de ressources personnalisés <code>Response</code>	le 20 novembre 2017	Vous pouvez désormais utiliser le champ <code>NoEcho</code> facultatif afin de masquer la sortie d'une ressource personnalisée. Pour plus d'informations, consultez Objets de réponse des ressources personnalisées. Le paramètre <code>noEcho</code> correspondant est pris en charge par la méthode <code>send</code>. Pour plus d'informations, consultez Module <code>cfn-response</code>.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Remplacements d'instances de piles ajoutés pour les ensembles de piles.	17 novembre	CloudFormation StackSets vous permet de remplacer les valeurs des paramètres dans les instances de pile par compte et par région. Vous pouvez remplacer les valeurs des paramètres lorsque vous créez les instances de piles, ou lors de la mise à jour des instances de piles existantes. Pour plus d'informations, consultez Paramètres de substitution sur des instances de piles .	2010-05-15
Ressource mise à jour	15 novembre	AWS::StepFunctions::StateMachine Vous pouvez utiliser <code>AWS::StepFunctions::StateMachine</code> pour spécifier un <code>StateMachineName</code> lors de la création d'une machine d'état. <code>DefinitionString</code> et <code>RoleArn</code> peuvent être mis à jour sans remplacer la machine d'état.	2010-05-15
StackSets prend désormais en charge un maximum de 500 instances de pile par ensemble de piles.	le 6 novembre 2010	Vous pouvez désormais créer un maximum de 500 instances de pile par ensemble de piles. Pour plus d'informations sur CloudFormation les limites, voir Comprendre CloudFormation les quotas .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	2 novembre 2010	AWS::CloudFront::CloudFrontOriginAccessIdentity Utilisez la <code>AWS::CloudFront::CloudFrontOriginAccessIdentity</code> ressource pour spécifier l'identité CloudFront d'accès à l'origine Amazon à associer à l'origine d'une CloudFront distribution. AWS::CloudFront::StreamingDistribution Utilisez la <code>AWS::CloudFront::StreamingDistribution</code> ressource pour spécifier une distribution de streaming RTMP (Adobe Real-Time Messaging Protocol) pour CloudFront.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	2 novembre 2010	AWS::ApiGateway::Deployment La propriété <code>StageName</code> est obsolète pour le type de propriété StageDescription. AWS::ApiGateway::Method La propriété <code>OperationName</code> permet d'attribuer un nom convivial à une méthode API Gateway. Utilisez la propriété <code>RequestValidatorId</code> pour associer un validateur de demande à une méthode. AWS::AutoScaling::AutoScalingGroup La propriété <code>LifecycleHookSpecificationList</code> permet de spécifier les actions à effectuer lorsque Auto Scaling lance ou résilie des instances. AWS::CloudFront::Distribution Utilisez la <code>Tags</code> propriété pour spécifier un ensemble arbitraire de balises (paires clé-valeur) à associer à une CloudFront distribution. Dans les types de DefaultCacheBehavior propriété CacheBehavioret, utilisez la <code>LambdaFunctionAssociations</code> propriété pour spécifier les associations de fonctions Lambda pour une CloudFront distribution. Dans le type de propriété CustomOriginConfig, la propriété <code>OriginKeepaliveTimeout</code> permet de spécifier un délai d'expiration personnalisé et la propriété <code>OriginReadTimeout</code> de spécifier un délai d'attente personnalisé des opérations de lecture à partir de l'origine.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		Dans le type de DistributionConfig propriété, utilisez la <code>IPV6Enabled</code> propriété pour indiquer s'il CloudFront répond aux requêtes IPv6 DNS avec une IPv6 adresse pour votre distribution. AWS::CodeDeploy::DeploymentGroup Dans le type de propriété LoadBalancerInfo, la propriété <code>TargetGroupInfoList</code> permet de spécifier des informations concernant un groupe cible dans Elastic Load Balancing à utiliser dans le cadre d'un déploiement. AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupEgress et AWS::EC2::SecurityGroupIngress Utilisez la propriété <code>Description</code> pour spécifier la description de la règle d'un groupe de sécurité. AWS::EC2::Subnet La propriété <code>Ipv6CidrBlock</code> prend désormais en charge les mises à jour <code>No interruption</code>. AWS::EC2::VPNGateway Utilisez la propriété <code>AmazonSideAsn</code> pour spécifier un ASN (Autonomous System Number) pour la partie Amazon d'une session BGP. AWS::EC2::VPNConnection Utilisez la propriété <code>VpnTunnelOptionsSpecifications</code> pour configurer les options de tunnel pour une connexion VPN.	

Modifier	Date de parution	Description	Version de l'API
		AWS::ElasticBeanstalk::ConfigurationTemplate et AWS::ElasticBeanstalk::Environment Dans les types de propriété ConfigurationOptionSetting et OptionSetting, utilisez la propriété <code>ResourceName</code> pour spécifier un nom de ressource pour une option de configuration de la mise à l'échelle basée sur la durée. AWS::EMR::Cluster Utilisez la propriété <code>CustomAmiId</code> pour spécifier une AMI Amazon Linux personnalisée pour un cluster. AWS::KinesisFirehose::DeliveryStream Utilisez l'attribut <code>Arn</code> avec la fonction <code>Fn::GetAtt</code> pour obtenir l'Amazon Resource Name (ARN) du flux de livraison. AWS::KMS::Key Utilisez la propriété <code>Tags</code> pour spécifier un ensemble arbitraire de balises (paires clé-valeur) à associer à une clé gérée par un client. AWS::OpsWorks::Layer et AWS::OpsWorks::Stack Utilisez la <code>Tags</code> propriété pour spécifier un ensemble arbitraire de balises (paires clé-valeur) à associer à une OpsWorks couche ou à une pile. AWS::RDS::OptionGroup Dans le type de propriété OptionConfiguration, la propriété <code>OptionVersion</code> permet de spécifier une version pour l'option.	

Modifier	Date de parution	Description	Version de l'API
		AWS::S3::Bucket La propriété <code>AnalyticsConfigurations</code> permet de configurer un filtre d'analyse pour un compartiment Amazon S3.	

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	24 octobre 20	AWS::Glue::Classifier La ressource <code>AWS::Glue::Classifier</code> permet de créer un processus de classification AWS Glue . AWS::Glue::Connection Utilisez la <code>AWS::Glue::Connection</code> ressource pour spécifier une AWS Glue connexion à une source de données. AWS::Glue::Crawler La ressource <code>AWS::Glue::Crawler</code> permet de spécifier un crawler AWS Glue . AWS::Glue::Database La ressource <code>AWS::Glue::Database</code> permet de créer une base de données AWS Glue . AWS::Glue::DevEndpoint Utilisez la ressource <code>AWS::Glue::DevEndpoint</code> pour spécifier un point de terminaison de développement pour le débogage à distance des scripts ETL. AWS::Glue::Job La ressource <code>AWS::Glue::Job</code> permet de spécifier une tâche AWS Glue dans le catalogue de données. AWS::Glue::Partition La ressource <code>AWS::Glue::Partition</code> permet de créer une partition AWS Glue , qui représente une tranche de données de table.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::Glue::Table La ressource <code>AWS::Glue::Table</code> permet de créer une table AWS Glue . AWS::Glue::Trigger Utilisez la <code>AWS::Glue::Trigger</code> ressource pour spécifier les déclencheurs qui exécutent les AWS Glue tâches.	
Nouvelles ressources	11 octobre 20	AWS::SSM::MaintenanceWindow Utilisez la <code>AWS::SSM::MaintenanceWindow</code> ressource pour créer une fenêtre AWS Systems Manager de maintenance. AWS::SSM::MaintenanceWindowTarget Utilisez la ressource <code>AWS::SSM::MaintenanceWindowTarget</code> pour enregistrer une cible avec une fenêtre de maintenance. AWS::SSM::MaintenanceWindowTask Utilisez la ressource <code>AWS::SSM::MaintenanceWindowTask</code> pour définir une tâche de fenêtre de maintenance. AWS::SSM::PatchBaseline La ressource <code>AWS::SSM::PatchBaseline</code> permet de définir une référence de correctifs Systems Manager.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	10 octobre 20	AWS::ElasticLoadBalancingV2::ListenerCertificate La ressource <code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code> permet de spécifier des certificats pour un écouteur Elastic Load Balancing.	2010-05-15
Nouvelle ressource	27 septembre	AWS::Athena::NamedQuery La ressource <code>AWS::Athena::NamedQuery</code> permet de créer une requête Amazon Athena.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	27 septembre	AWS::EC2::NatGateway Utilisez la propriété <code>Tags</code> afin de spécifier des balises de ressource pour une passerelle NAT. AWS::ElasticBeanstalk::Application La propriété <code>ResourceLifecycleConfig</code> permet de définir les paramètres de cycle de vie des ressources qui appartiennent à l'application ainsi que le rôle de service qu'endosse Elastic Beanstalk afin d'appliquer les paramètres de cycle de vie. AWS::ElasticBeanstalk::ConfigurationTemplate et AWS::ElasticBeanstalk::Environment La propriété <code>PlatformArn</code> permet de spécifier une plateforme personnalisée pour Elastic Beanstalk. AWS::ElasticLoadBalancingV2::TargetGroup Dans le type de TargetDescription propriété, utilisez la <code>AvailabilityZone</code> propriété pour spécifier la zone de disponibilité dans laquelle l'adresse IP doit être enregistrée. AWS::Events::Rule Dans le type de propriété Target, utilisez les propriétés suivantes pour la transformation d'entrée des événements et la définition des cibles de tâche Amazon ECS et de flux Kinesis. • <code>EcsParameters</code> • <code>InputTransformer</code> • <code>KinesisParameters</code> • <code>RunCommandParameters</code>	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::KinesisFirehose::DeliveryStream Utilisez la <code>DeliveryStreamType</code> propriété pour spécifier le type de flux et la <code>KinesisStreamSourceConfiguration</code> propriété pour spécifier le flux et le rôle ARNs d'un flux Kinesis utilisé comme source pour un flux de diffusion. AWS::RDS::DBInstance Pour la propriété <code>Engine</code>, si vous avez spécifié <code>oracle-se</code> ou <code>oracle-se1</code>, vous pouvez mettre à jour vers <code>oracle-se2</code> sans que l'instance de base de données ne soit remplacée. AWS::S3::Bucket La propriété <code>AccelerateConfiguration</code> permet de configurer l'état de la fonction <code>Transfer Acceleration</code> pour un compartiment Amazon S3.	
Ajout de la protection contre la résiliation pour les piles.	26 septembre 2017	L'activation de la protection de la résiliation sur une pile empêche la suppression accidentelle de celle-ci. Un utilisateur ne peut pas supprimer une pile pour laquelle la protection contre la résiliation est activée. Pour plus d'informations, consultez Protection d'une pile contre la suppression .	2010-05-15
Modification de la valeur par défaut <code>umask</code> à partir de la version 1.4-22	14 septembre 2017	La valeur du paramètre <code>umask</code> par défaut pour le fichier de configuration <code>cfn-hup.conf</code> est désormais <code>022</code> . Pour plus d'informations, consultez cfn-hup .	

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	7 septembre 2017	AWS::ElasticLoadBalancingV2::LoadBalancer Utilisez la <code>SubnetMappings</code> propriété pour spécifier les IDs sous-réseaux à associer à l'équilibreur de charge. Utilisez la propriété <code>Type</code> pour spécifier le type d'équilibreur de charge à créer. AWS::ElasticLoadBalancingV2::TargetGroup Utilisez la propriété <code>TargetType</code> pour spécifier le type d'enregistrement des cibles du groupe cible.	2010-05-15
Des déclencheurs de rollback ont été ajoutés à l'API CloudFormation	31 août 2017	Les déclencheurs d'annulation vous permettent de CloudFormation surveiller l'état de votre application lors de la création et de la mise à jour de la pile, et d'annuler cette opération si l'application dépasse le seuil de l'une des alarmes que vous avez spécifiées. Pour plus d'informations, consultez RollbackConfiguration dans la Référence d'API CloudFormation .	2010-05-15
Nouveau paramètre <code>umask</code> pour le fichier <code>cfn-hup.conf</code>	31 août 2017	Utilisez le paramètre <code>umask</code> du fichier de configuration <code>cfn-hup.conf</code> pour contrôler les autorisations de fichiers utilisées par le démon <code>cfn-hup</code> (version 1.4-21). Pour plus d'informations, consultez cfn-hup .	

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour pour la prise en charge du dimensionnement d'un VPC	29 août 2017	AWS::EC2::VPCCidrBlock Utilisez la <code>CidrBlock</code> propriété pour associer un bloc IPv4 CIDR à un VPC. AWS::EC2::VPC Utilisez l' <code>CidrBlockAssociations</code> attribut associé à la <code>Fn::GetAtt</code> fonction pour obtenir une liste des associations de blocs IPv4 CIDR IDs associées au VPC.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	23 août 2017	AWS::S3::Bucket Dans le type de propriété Rule, la propriété <code>TagFilters</code> permet de spécifier les balises à utiliser pour identifier un sous-ensemble d'objets pour un compartiment Amazon S3. Utilisez la <code>MetricsConfiguration</code> propriété pour spécifier une configuration de métriques pour les métriques de CloudWatch demande provenant d'un compartiment Amazon S3. AWS::IoT::TopicRule Dans le type de propriété Action, la propriété <code>DynamoDBv2Action</code> permet de décrire une action AWS IoT qui écrit des données dans une table DynamoDB. Dans le type de propriété Action, la propriété <code>DynamoDBAction</code> prend désormais en charge les propriétés <code>HashKeyType</code> et <code>RangeKeyType</code> . AWS::Lambda::Permission Utilisez la propriété <code>EventSourceToken</code> pour spécifier un jeton unique qui doit être fourni par le principal qui appelle la fonction.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouveaux pseudo-paramètres	23 août 2017	Utilisez le pseudo-paramètre <code>AWS::Partition</code> pour renvoyer la partition qui contient une ressource. Utilisez le pseudo-paramètre <code>AWS::URLSuffix</code> pour retourner le suffixe pour un domaine. Pour plus d'informations, consultez Référence des pseudo-paramètres.	2010-05-15
Nouvelles ressources pour la prise en charge DAX	22 août 2017	AWS::DAX::Cluster La ressource <code>AWS::DAX::Cluster</code> permet de créer un cluster DAX à utiliser avec Amazon DynamoDB. AWS::DAX::ParameterGroup La ressource <code>AWS::DAX::ParameterGroup</code> permet de créer un groupe de paramètres à utiliser avec Amazon DynamoDB. AWS::DAX::SubnetGroup Utilisez la ressource <code>AWS::DAX::SubnetGroup</code> pour créer un groupe de sous-réseau à utiliser avec DAX (DynamoDB Accelerator).	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	18 août 2017	AWS::ApiGateway::DocumentationPart et AWS::ApiGateway::DocumentationVersion permettent de créer une documentation pour votre API API Gateway. AWS::ApiGateway::GatewayResponse La ressource <code>AWS::ApiGateway::GatewayResponse</code> permet de créer une réponse personnalisée pour votre API API Gateway. AWS::ApiGateway::RequestValidator La ressource <code>AWS::ApiGateway::RequestValidator</code> permet de configurer des règles de validation pour les demandes entrantes à destination de votre API API Gateway. AWS::EC2::NetworkInterfacePermission Utilisez la <code>AWS::EC2::NetworkInterfacePermission</code> ressource pour autoriser un AWS compte à accéder à une interface réseau.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	18 août 2017	AWS::ApiGateway::Stage Utilisez la propriété <code>DocumentationVersion</code> pour spécifier un instantané versionné de la documentation de l'API. AWS::AutoScaling::ScalingPolicy La propriété <code>TargetTrackingConfiguration</code> permet de spécifier une configuration de politique de mise à l'échelle Suivi de cible Auto Scaling. AWS::CloudTrail::Trail Utilisez la propriété <code>EventSelectors</code> pour la prise en charge des événements de données Amazon S3. AWS::CodeDeploy::DeploymentGroup Les propriétés <code>LoadBalancerInfo</code> et <code>DeploymentStyle</code> permettent de spécifier un équilibreur de charge Elastic Load Balancing pour un déploiement sur place. Utilisez la propriété <code>AutoRollbackConfiguration</code> afin de configurer une restauration automatique pour le déploiement. AWS::EC2::SpotFleet Dans le type de propriété SpotFleetRequestConfigurationData, la propriété <code>ReplaceUnhealthyInstances</code> permet d'indiquer si le parc d'instances Spot doit remplacer les instances non saines et la propriété <code>Type</code> de spécifier le type de demande.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::EC2::Subnet Utilisez les <code>Ipv6CidrBlock</code> propriétés <code>AssignIpv6AddressOnCreation</code> et pour créer un sous-réseau avec un bloc IPv6 CIDR. AWS::KinesisFirehose::DeliveryStream La propriété <code>ExtendedS3DestinationConfiguration</code> permet de configurer une destination dans Amazon S3. Utilisez la sous-propriété <code>ProcessingConfiguration</code> dans chaque configuration de destination pour appeler les fonctions Lambda qui transforment les données source entrantes et diffuser les données transformées à leurs destinations. AWS::RDS::DBCluster et AWS::RDS::DBInstance La <code>DeletionPolicy</code> par défaut est désormais <code>Snapshot</code> pour certains types de ressources <code>AWS::RDS::DBCluster</code> ainsi que pour les ressources <code>AWS::RDS::DBInstance</code> qui ne spécifient pas explicitement la propriété <code>DBClusterIdentifier</code>. Pour plus d'informations, consultez Attribut DeletionPolicy. AWS::S3::Bucket Dans le type de propriété Rule, la propriété <code>AbortIncompleteMultipartUpload</code> permet de spécifier une règle de cycle de vie qui annule les chargements partitionnés incomplets dans un compartiment Amazon S3.	

Modifier	Date de parution	Description	Version de l'API
		AWS::SQS::Queue Les propriétés <code>KmsMasterKeyId</code> et <code>KmsDataKeyReusePeriodSeconds</code> permettent de configurer un chiffrement côté serveur pour Amazon SQS. Ajout de l'attribut <code>Arn</code> à la fonction intrinsèque <code>Fn::GetAtt</code> pour les ressources suivantes : • AWS::CloudTrail::Trail. Autre ajout : <code>SnsTopicArn</code> . • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	
Support pour les tags de pile dans les CodePipeline artefacts	18 août 2017	Vous pouvez désormais spécifier des balises pour les piles dans les fichiers de configuration des modèles afin de les utiliser comme artefacts pour les CodePipeline pipelines. Les balises spécifiées sont appliquées à des piles créées à l'aide du fichier de configuration de modèle. Pour plus d'informations, consultez CloudFormation Artefacts.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Créez des systèmes de fichiers chiffrés.	14 août 2017	AWS::EFS::FileSystem La propriété <code>Encrypted</code> permet de chiffrer un système de fichiers Amazon EFS pendant la création. Utilisez la propriété <code>KmsKeyId</code> pour, si vous le souhaitez, indiquer une clé gérée par un client personnalisée à utiliser pour protéger le système de fichiers chiffré.	2010-05-15
Nouvelles ressources d'AWS Batch assistance	8 août 2017	AWS::Batch::ComputeEnvironment Utilisez la <code>AWS::Batch::ComputeEnvironment</code> ressource pour définir votre environnement AWS Batch informatique. AWS::Batch::JobDefinition Utilisez la <code>AWS::Batch::JobDefinition</code> ressource pour spécifier les paramètres d'une définition de AWS Batch tâche. AWS::Batch::JobQueue Utilisez la <code>AWS::Batch::JobQueue</code> ressource pour définir votre file AWS Batch d'attente de tâches.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources pour le support du service géré Amazon pour Apache Flink	28 juillet 2017	AWS::KinesisAnalytics::Application Utilisez la ressource <code>AWS::KinesisAnalytics::Application</code> pour créer une application de service géré Amazon pour Apache Flink. AWS::KinesisAnalytics::ApplicationOutput Utilisez la ressource <code>AWS::KinesisAnalytics::ApplicationOutput</code> pour ajouter une destination externe à votre application de service géré Amazon pour Apache Flink. AWS::KinesisAnalytics::ApplicationReferenceDataSource Utilisez la ressource <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code> pour ajouter une source de données de référence à une application de service géré Amazon pour Apache Flink existante.	2010-05-15
StackSets À utiliser pour gérer de manière centralisée les stacks entre les comptes et les régions	25 juillet 2017	StackSets vous permet de créer, de mettre à jour ou de supprimer des piles sur plusieurs comptes et régions en une seule opération. À l'aide d'un compte administrateur, vous définissez et gérez un CloudFormation modèle, et vous l'utilisez comme base pour provisionner des piles dans des comptes cibles sélectionnés dans des régions spécifiques. Pour plus d'informations, consultez la section Gestion des piles entre les comptes et les régions avec StackSets.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Afficher les événements de pile par jeton de demande de client	14 juillet 2017	Dans la console, les opérations de pile affichent le jeton de demande de client dans l'onglet Événements. Tous les événements déclenchés par une opération de pile donnée reçoivent le même jeton de demande de client, lequel peut être utilisé pour suivre les opérations. Pour plus d'informations, consultez la section Affichage des données et des ressources de la CloudFormation pile sur AWS Management Console et StackEvent dans le guide de référence des AWS CloudFormation API.	2010-05-15
Utiliser les liens de création rapide de piles	14 juillet 2017	Utilisez les liens de création rapide de piles pour obtenir des piles rapidement opérationnelles. Vous pouvez spécifier l'URL du modèle, le nom de la pile et les paramètres du modèle pour préremplir une seule page Create Stack Wizard (Assistant Créer une pile). Pour plus d'informations, consultez Création de liens de création rapide pour piles .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources d'AWS Database assistance	12 juillet 2017	AWS::DMS::Certificate Utilisez la <code>AWS::DMS::Certificate</code> ressource pour créer un certificat SSL qui chiffre les connexions entre les AWS DMS points de terminaison et l'instance de réplication. AWS::DMS::Endpoint La ressource <code>AWS::DMS::Endpoint</code> permet de créer un point de terminaison AWS DMS . AWS::DMS::EventSubscription Utilisez cette <code>AWS::DMS::EventSubscription</code> ressource pour recevoir des notifications d' AWS DMS événements via Amazon Simple Notification Service. AWS::DMS::ReplicationInstance Utilisez la <code>AWS::DMS::ReplicationInstance</code> ressource pour créer une instance AWS DMS de réplication. AWS::DMS::ReplicationSubnetGroup Utilisez la <code>AWS::DMS::ReplicationSubnetGroup</code> ressource pour créer un groupe de sous-réseaux de AWS DMS réplication. AWS::DMS::ReplicationTask Utilisez la <code>AWS::DMS::ReplicationTask</code> ressource pour créer une tâche AWS DMS de réplication.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	5 juillet 2017	AWS::CloudWatch::Dashboard Utilisez cette <code>AWS::CloudWatch::Dashboard</code> ressource pour définir un CloudWatch tableau de bord personnalisé pour votre CloudWatch console. AWS::ApiGateway::DomainName La ressource <code>AWS::ApiGateway::DomainName</code> permet de spécifier une URL conviviale personnalisée pour votre API qui est déployée dans Amazon API Gateway. AWS::EC2::EgressOnlyInternetGateway Utilisez la ressource <code>AWS::EC2::EgressOnlyInternetGateway</code> pour créer une passerelle Internet de sortie uniquement pour votre VPC. InstanceFleetConfig La ressource <code>InstanceFleetConfig</code> permet de configurer un parc d'instances Spot pour un cluster Amazon EMR.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	5 juillet 2017	AWS::ApiGateway::RestApi Utilisez la propriété <code>BinaryMediaTypes</code> pour spécifier les types de supports binaires pris en charge. AWS::ApplicationAutoScaling::ScalingPolicy Utilisez la propriété <code>TargetTrackingScalingPolicyConfiguration</code> pour spécifier une configuration de politique de suivi des objectifs et d'échelonnement. AWS::CloudTrail::Trail Utilisez la <code>TrailName</code> propriété pour spécifier un nom personnalisé pour une AWS CloudTrail ressource. Utilisez la propriété <code>Tags</code> pour spécifier des balises de ressource. AWS::CodeDeploy::DeploymentGroup Utilisez la propriété <code>AlarmConfiguration</code> afin de configurer des alarmes pour un groupe de déploiement. Utilisez la propriété <code>TriggerConfigurations</code> afin de configurer des déclencheurs de notification pour un groupe de déploiement. AWS::EMR::Cluster Utilisez la <code>CoreInstanceFleet</code> propriété et la <code>MasterInstanceFleet</code> propriété du type de JobFlowInstancesConfig propriété pour configurer le parc d'instances Spot pour un cluster Amazon EMR.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::DynamoDB::Table La propriété <code>TimeToLiveSpecification</code> permet de spécifier les paramètres de durée de vie (TTL) pour une table Amazon DynamoDB. La propriété <code>Tags</code> permet de spécifier des balises de ressource pour une table DynamoDB. AWS::EC2::Instance La propriété <code>IamInstanceProfile</code> prend désormais en charge les mises à jour <code>No interruption</code>. AWS::EC2::Route Utilisez la propriété <code>EgressOnlyInternetGatewayId</code> pour spécifier une passerelle Internet de sortie uniquement pour un routage EC2. AWS::Kinesis::Stream Utilisez la propriété <code>RetentionPeriodHours</code> afin de spécifier le nombre d'heures pendant lesquelles des enregistrements de données stockés dans des partitions restent accessibles. AWS::RDS::DBCluster La propriété <code>ReplicationSourceIdentifier</code> permet de créer un cluster de base de données en tant que réplica en lecture d'un autre cluster de base de données ou d'une instance de base de données MySQL Amazon RDS.	

Modifier	Date de parution	Description	Version de l'API
		AWS::Redshift::Cluster La propriété <code>LoggingProperties</code> permet de créer des fichiers journaux d'audit et de les stocker dans Amazon S3.	
Nouvelles ressources	6 juin 2017	AWS::EMR::SecurityConfiguration Utilisez la ressource <code>AWS::EMR::SecurityConfiguration</code> pour créer une configuration de sécurité qui est stockée dans le service et peut être spécifiée quand un cluster est créé.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	6 juin 2017	AWS::AutoScaling::LifecycleHook Les propriétés <code>NotificationTargetARN</code> et <code>RoleARN</code> sont désormais facultatives. AWS::CloudWatch::Alarm Vous pouvez désormais utiliser les propriétés <code>EvaluateLowSampleCountPercentile</code>, <code>ExtendedStatistic</code> et <code>TreatMissingData</code> lors de la création de ressources <code>AWS::CloudWatch::Alarm</code>. AWS::EC2::SpotFleet CloudFormation prend en charge les modifications modifiables des propriétés du parc Spot. Les propriétés suivantes de la propriété <code>SpotFleetRequestConfigData</code> prennent en charge les mises à jour <code>Replacement</code> : • <code>AllocationStrategy</code> • <code>IamFleetRole</code> • <code>LaunchSpecifications</code> • <code>SpotPrice</code> • <code>TerminateInstancesWithExpiration</code> • <code>ValidFrom</code> • <code>ValidUntil</code> Les propriétés suivantes de la propriété <code>SpotFleetRequestConfigData</code> prennent en charge les mises à jour <code>No interruption</code> : • <code>ExcessCapacityTerminationPolicy</code> • <code>TargetCapacity</code>	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::EMR::InstanceGroupConfig CloudFormation prend désormais en charge les groupes d'instances de tâches Auto Scaling pour Amazon EMR. AWS::Events::Rule La propriété RoleArn est obsolète sur la ressource Rule. La propriété RoleArn du type de propriété Target permet de spécifier le rôle IAM à utiliser pour une cible. AWS::Kinesis::Stream La propriété ShardCount prend désormais en charge les mises à jour No interruption . AWS::Lambda::Function La propriété TracingConfig permet de configurer les paramètres de suivi pour les fonctions Lambda. AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, et AWS::Redshift::ClusterSubnetGroup Utilisez la propriété Tags pour spécifier des balises de ressource. AWS::RDS::DBCluster Ajout de l'attribut ReadEndpoint.Address à la fonction intrinsèque Fn::GetAtt .	

Modifier	Date de parution	Description	Version de l'API
		AWS::S3::Bucket Ajout de l'attribut Arn à la fonction intrinsèque Fn::GetAtt .	

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	11 mai 2017	Les nouvelles ressources suivantes prennent en charge l'utilisation AWS WAF des équilibreurs de charge d'application Elastic Load Balancing (ELB). AWS::WAFRegional::ByteMatchSet Utilisez la ressource <code>AWS::WAFRegional::ByteMatchSet</code> pour identifier une partie d'une demande web que vous voulez inspecter. AWS::WAFRegional::IPSet Utilisez la ressource <code>AWS::WAFRegional::IPSet</code> pour spécifier les demandes web à autoriser ou à bloquer en fonction de leur adresse IP d'origine. AWS::WAFRegional::Rule Utilisez la ressource <code>AWS::WAFRegional::Rule</code> pour spécifier une combinaison d'objets <code>IPSet</code>, <code>ByteMatchSet</code> et <code>SqlInjectionMatchSet</code>, qui identifient les demandes web à autoriser, bloquer ou comptabiliser. AWS::WAFRegional::SizeConstraintSet Utilisez la ressource <code>AWS::WAFRegional::SizeConstraintSet</code> pour spécifier une contrainte de taille utilisée pour vérifier la taille de différentes parties d'une demande web et définir quelles parties inspecter. AWS::WAFRegional::SqlInjectionMatchSet Utilisez la ressource <code>AWS::WAFRegional::SqlInjectionMatchSet</code> pour autoriser, bloquer ou compter les demandes web dont une partie spécifique contient du code SQL malveillant.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::WAFRegional::WebACL Utilisez la ressource <code>AWS::WAFRegional::WebACL</code> pour identifier les demandes web à autoriser, bloquer ou compter.	
		AWS::WAFRegional::WebACLAssociation Utilisez la ressource <code>AWS::WAFRegional::WebACLAssociation</code> pour associer une liste de contrôle d'accès (ACL) web à une ressource.	
		AWS::WAFRegional::XssMatchSet Utilisez la ressource <code>AWS::WAFRegional::XssMatchSet</code> pour spécifier les parties d'une demande web dans lesquelles AWS WAF doit rechercher les attaques de scripts intersites et inspecter le nom de l'en-tête.	

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	28 avril 2017	<u>AWS::Cognito::IdentityPool</u> La ressource <code>AWS::Cognito::IdentityPool</code> permet de créer un groupe d'identités Amazon Cognito. <u>AWS::Cognito::IdentityPoolRoleAttachment</u> La ressource <code>AWS::Cognito::IdentityPoolRoleAttachment</code> permet de gérer la configuration de rôle pour un groupe d'identités Amazon Cognito. <u>AWS::Cognito::UserPool</u> La ressource <code>AWS::Cognito::UserPool</code> permet de créer un groupe d'utilisateurs Amazon Cognito. <u>AWS::Cognito::UserPoolClient</u> Utilisez la ressource <code>AWS::Cognito::UserPoolClient</code> pour créer un client de groupe d'utilisateurs. <u>AWS::Cognito::UserPoolGroup</u> La ressource <code>AWS::Cognito::UserPoolGroup</code> permet de créer un ensemble d'utilisateurs dans un groupe d'utilisateurs Amazon Cognito. <u>AWS::Cognito::UserPoolUser</u> La ressource <code>AWS::Cognito::UserPoolUser</code> permet de créer un utilisateur de groupe d'utilisateurs Amazon Cognito. <u>AWS::Cognito::UserPoolUserToGroupAttachment</u> La ressource <code>AWS::Cognito::UserPoolUserToGroupAttachment</code> permet d'attacher un utilisateur	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		ur à un ensemble d'utilisateurs d'un groupe d'utilisateurs Amazon Cognito.	

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	28 avril 2017	SourceDetails Utilisez la sous-propriété <code>MaximumExecutionFrequency</code> de la ressource <code>AWS::Config::ConfigRule</code> afin d'exécuter des évaluations pour une règle personnalisée à l'aide d'un déclencheur périodique. AWS::EC2::Volume Nous prenons désormais en charge Elastic Volumes pour Amazon Elastic Block Store (Amazon EBS) dans CloudFormation. Nous prenons désormais en charge les mises à jour <code>No interruption</code> sur trois propriétés : <code>VolumeType</code> , <code>Size</code> et <code>Iops</code>. AWS::EC2::SecurityGroup La propriété <code>GroupName</code> permet de spécifier un nom pour un groupe de sécurité Amazon EC2. AWS::ECS::Service Il existe trois nouvelles propriétés pour <code>AWS::ECS::Service</code> : <code>PlacementConstraints</code> , <code>PlacementStrategies</code> et <code>ServiceName</code> . AWS::ECS::TaskDefinition Utilisez la propriété <code>PlacementConstraints</code> afin de définir des contraintes de placement pour les tâches dans le service. AWS::ElastiCache::ReplicationGroup Ajout de l'attribut <code>ConfigurationEndpoint.Address</code> et de l'attribut <code>Configura</code>	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		<code>tionEndPoint.Port</code> à la fonction intrinsèque <code>Fn::GetAtt</code> . AWS::ElasticLoadBalancingV2::LoadBalancer Utilisez la propriété <code>IpAddressType</code> pour spécifier le type des adresses IP utilisées par les sous-réseaux de l'équilibreur de charge. AWS::EMR::Cluster CloudFormation prend désormais en charge Auto Scaling pour les clusters Amazon EMR. AWS::IAM::ManagedPolicy La propriété <code>ManagedPolicyName</code> permet de spécifier un nom personnalisé pour votre politique gérée IAM. AWS::Lambda::Function La propriété <code>Tags</code> permet d'ajouter des balises à votre fonction Lambda. AWS::OpsWorks::Instance Ajout des attributs suivants à la fonction intrinsèque <code>Fn::GetAtt</code> : <code>AvailabilityZone</code> , <code>PrivateDnsName</code> , <code>PrivateIp</code> et <code>PublicDnsName</code> . AWS::OpsWorks::UserProfile Utilisez la propriété <code>SshUsername</code> pour spécifier le nom SSH d'un utilisateur. Ajout de l'attribut <code>SshUsername</code> à la fonction intrinsèque <code>Fn::GetAtt</code> .	

Modifier	Date de parution	Description	Version de l'API
		AWS::Redshift::Cluster Utilisez la <code>IamRoles</code> propriété pour fournir une liste d'un ou de plusieurs Gestion des identités et des accès AWS rôles que le cluster Amazon Redshift peut utiliser pour accéder à d'autres AWS services.	
Modifier des modèles en YAML et JSON à l'aide CloudFormation de Designer	6 avril 2017	Lorsque vous créez des CloudFormation modèles à l'aide de Designer, vous pouvez désormais modifier votre modèle en YAML et en JSON dans l'éditeur intégré. Vous pouvez également convertir des modèles JSON en YAML et inversement, en fonction de votre langage de création préféré. Pour plus d'informations, voir Qu'est-ce que CloudFormation Designer ?.	2010-05-15
Nouvelle ressource	6 avril 2017	AWS::SSM::Parameter La ressource <code>AWS::SSM::Parameter</code> permet de créer un paramètre SSM dans Parameter Store.	2010-05-15
Transformation <code>AWS::Include</code>	28 mars 2017	La transformation <code>AWS::Include</code> permet de référencer des extraits réutilisables stockés dans un compartiment Amazon S3. Pour plus d'informations, consultez la section Transformation AWS::Include (français non garanti).	2010-05-15
Apparez votre Amazon VPC avec un autre compte	28 mars 2017	Vous pouvez désormais associer votre Amazon VPC CloudFormation à un VPC d'un autre compte. AWS Pour plus d'informations, consultez Peer avec un Amazon VPC dans un autre AWS compte.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	28 mars 2017	AWS::ApiGateway::UsagePlanKey Utilisez la ressource <code>AWS::ApiGateway::UsagePlanKey</code> pour associer une clé de plan d'utilisation et identifier les utilisateurs auxquels le plan d'utilisation est appliqué.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	28 mars 2017	AWS::EC2::VPCPeeringConnection La propriété <code>PeerOwnerId</code> et la propriété <code>PeerRoleArn</code> permettent de s'appairer avec un VPC d'un autre compte AWS . Pour plus d'informations, consultez Peer avec un Amazon VPC dans un autre AWS compte. AWS::IAM::InstanceProfile Utilisez la propriété <code>InstanceProfileName</code> pour configurer un profil d'instance. AWS::Lambda::Function La propriété <code>DeadLetterConfig</code> permet de configurer la façon dont AWS Lambda gère les événements qu'il ne peut pas traiter. Node.js v0.10 n'est plus pris en charge pour la propriété <code>Runtime</code>. AWS::Route53::HealthCheck Sept nouveaux types de sous-propriété de ressource ont été ajoutés pour la propriété HealthCheckConfig : <code>AlarmIdentifier</code> , <code>ChildHealthChecks</code> , <code>EnableSNI</code> , <code>HealthThreshold</code> , <code>InsufficientDataHealthStatus</code> , <code>Inverted</code> et <code>MeasureLatency</code> . AWS::SQS::Queue Utilisez les <code>FifoQueue</code> propriétés <code>ContentBasedDeduplication</code> et pour créer des files d'attente Amazon Simple Queue Service First-In-First-Out (FIFO).	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::S3::Bucket Vous pouvez désormais spécifier des noms de IPv6 domaine pour vos compartiments Amazon S3.	
Nouvelles ressources	10 février 201	AWS::StepFunctions::Activity Utilisez la <code>AWS::StepFunctions::Activity</code> ressource pour créer une AWS Step Functions activité. AWS::StepFunctions::StateMachine Utilisez la ressource <code>AWS::StepFunctions::StateMachine</code> pour créer une machine d'état Step Functions.	2010-05-15
Nouvelle fonction intrinsèque	17 janvier 201	Utilisez la fonction <code>Fn::Split</code> pour fractionner une chaîne en liste de valeurs de chaîne. Pour de plus amples informations, veuillez consulter Fn::Split .	2010-05-15
Prise en charge des importations de listes par la console	17 janvier 201	Utilisez la CloudFormation console pour voir toutes les piles qui importent une valeur de sortie exportée. Pour plus d'informations, consultez Affichage des piles qui importent une valeur de sortie exportée .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	17 janvier 201	AWS::AutoScaling::AutoScalingGroup La propriété <code>LoadBalancerNames</code> peut être mise à jour sans remplacer le groupe Auto Scaling. AWS::ECS::TaskDefinition Ajout des propriétés <code>NetworkMode</code> et <code>MemoryReservation</code> . AWS::RDS::DBCluster CloudFormation prend en charge les mises à jour de la <code>Tags</code> propriété. AWS::RDS::DBInstance Ajout de la propriété <code>Timezone</code>. FirehoseAction Ajout de la propriété <code>Separator</code> . AWS::OpsWorks::Instance Ajout de l'attribut <code>PublicIp</code> pour la fonction intrinsèque <code>Fn::GetAtt</code> .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	01 décembre 2016	AWS::CodeBuild::Project Utilisez cette <code>AWS::CodeBuild::Project</code> ressource pour créer un AWS CodeBuild projet qui définit le mode de CodeBuild construction de votre code source. AWS::SSM::Association La ressource <code>AWS::SSM::Association</code> permet d'associer un document Amazon EC2 Systems Manager à des instances EC2. AWS::EC2::SubnetCidrBlock Utilisez la <code>AWS::EC2::SubnetCidrBlock</code> ressource pour associer un seul bloc IPv6 CIDR à un sous-réseau Amazon VPC. AWS::EC2::VPCCidrBlock Utilisez la <code>AWS::EC2::VPCCidrBlock</code> ressource pour associer un seul bloc IPv6 CIDR fourni par Amazon à un Amazon VPC.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources d'IPv6 assistance mises à jour	01 décembre 2016	AWS::EC2::Instance Ajout des propriétés <code>Ipv6AddressCount</code> et <code>Ipv6Addresses</code> . AWS::EC2::NetworkAclEntry Ajout de la propriété <code>Ipv6CidrBlock</code> . AWS::EC2::NetworkInterface Ajout des propriétés <code>Ipv6AddressCount</code> et <code>Ipv6Addresses</code> . AWS::EC2::Route Ajout de la propriété <code>DestinationIpv6CidrBlock</code> . AWS::EC2::SecurityGroupEgress Ajout de la propriété <code>CidrIpv6</code>. AWS::EC2::SecurityGroupIngress Ajout de la propriété <code>CidrIpv6</code>. AWS::EC2::SpotFleet Ajout des propriétés <code>Ipv6AddressCount</code> et <code>Ipv6Addresses</code> pour les interfaces réseau de spécification de lancement. AWS::EC2::Subnet Ajout de l'attribut <code>Ipv6CidrBlocks</code> pour la fonction <code>Fn::GetAtt</code> . AWS::EC2::VPC Ajout de l'attribut <code>Ipv6CidrBlocks</code> pour la fonction <code>Fn::GetAtt</code> .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::SSM::Document Ajout de la propriété DocumentType .	
Spécification de ressource	22 novembre	Utilisez la spécification CloudFormation des ressources pour créer des outils qui vous aideront à créer des CloudFormation modèles. La spécification est un fichier texte au format JSON, lisible par la machine. Pour plus d'informations, consultez la section Spécification CloudFormation des ressources .	2010-05-15
Nouvelles ressources	22 novembre	AWS::OpsWorks::UserProfile Utilisez la <code>AWS::OpsWorks::UserProfile</code> ressource pour configurer l'accès SSH pour les utilisateurs qui ont besoin d'accéder aux instances d'une OpsWorks pile. AWS::OpsWorks::Volume Utilisez cette <code>AWS::OpsWorks::Volume</code> ressource pour enregistrer un volume Amazon Elastic Block Store auprès d'une OpsWorks pile.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	22 novembre	AWS::OpsWorks::App Ajout de la propriété <code>DataSources</code> . AWS::OpsWorks::Instance Ajout des propriétés <code>BlockDeviceMappings</code> , <code>AgentVersion</code> , <code>ElasticIps</code> , <code>Hostname</code> , <code>Tenancy</code> et <code>Volumes</code>. AWS::OpsWorks::Layer Ajout des propriétés <code>CustomJson</code> et <code>VolumeConfigurations</code> . AWS::OpsWorks::Stack Ajout des propriétés <code>ElasticIps</code> , <code>EcsClusterArn</code> , <code>RdsDbInstances</code> , <code>CloneAppIds</code> , <code>ClonePermissions</code> et <code>SourceStackId</code> . AWS::RDS::DBInstance Ajout de la propriété <code>CopyTagsToSnapshot</code> .	2010-05-15
Liste des importations	22 novembre	Répertoriez les importations d'une valeur de sortie exportée pour CloudFormation savoir quelles piles importent la valeur. Pour plus d'informations, consultez Affichage des piles qui importent une valeur de sortie exportée.	2010-05-15
Transforme	17 novembre	Spécifiez le modèle d'application AWS sans serveur (AWS SAM) CloudFormation utilisé pour traiter la syntaxe AWS SAM pour les applications sans serveur. Pour plus d'informations, consultez la section Transformation (français non garanti).	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	17 novembre	AWS::SNS::Subscription La ressource <code>AWS::SNS::Subscription</code> permet d'abonner un point de terminaison à une rubrique Amazon Simple Notification Service.	2010-05-15
Ressource mise à jour	17 novembre	AWS::Lambda::Function Utilisez la <code>Environment</code> propriété pour spécifier des paires clé-valeur (variables d'environnement) auxquelles votre AWS Lambda fonction peut accéder. Utilisez la <code>KmsKeyArn</code> propriété pour spécifier une clé KMS AWS Lambda utilisée pour chiffrer et déchiffrer les variables d'environnement.	2010-05-15
Nouvelles commandes CLI	17 novembre	Chargement d'artefacts locaux dans un compartiment S3 Utilisez la package commande pour télécharger les artefacts locaux référencés dans un CloudFormation modèle vers un compartiment S3. Déploiement rapide de modèles avec des transformations Utilisez la commande <code>deploy</code> pour associer les actions de jeux de modifications Créer et Exécuter dans une seule commande. Cette commande est utile pour créer ou mettre à jour rapidement des piles contenant des transformations.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	03 novembre	AWS::CloudFront::Distribution Pour la DistributionConfig propriété, utilisez-la <code>HttpVersion</code> pour spécifier la dernière version HTTP que les utilisateurs peuvent utiliser pour communiquer avec Amazon CloudFront. Pour la ForwardedValues propriété, utilisez-la <code>QueryStringCacheKeys</code> pour spécifier les paramètres de chaîne de requête CloudFront utilisés pour déterminer le contenu à mettre en cache.	2010-05-15
Liste des exportations de pile	03 novembre	Utilisez la CloudFormation console, l'API ou AWS CLI pour consulter la liste de toutes les valeurs de sortie exportées pour une région. Pour plus d'informations, consultez Exportation de valeurs de sortie de pile .	2010-05-15
Livraison en continu avec des piles	03 novembre	AWS CodePipeline À utiliser pour créer des flux de livraison continus avec des CloudFormation piles. Pour plus d'informations, consultez la section Livraison continue avec CodePipeline .	2010-05-15
Ignorer des ressources pendant la restauration	03 novembre	Si une pile présente l'état <code>UPDATE_ROLLBACK_FAILED</code> , utilisez le paramètre <code>ResourcesToSkip</code> pour l'action <code>ContinueUpdateRollback</code> pour ignorer les ressources qui ne peuvent pas être restaurées par CloudFormation. Pour plus d'informations, consultez la section Dépannage dans Échec de la restauration de la mise à jour .	2010-05-15
Amélioration des jeux de modifications	03 novembre	Vous pouvez créer une nouvelle pile à l'aide d'un jeu de modifications .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	12 octobre 20	AWS::ElastiCache::CacheCluster Mettez à jour la propriété <code>CacheNodeType</code> sans remplacer le cluster. AWS::ElastiCache::ReplicationGroup Vous pouvez créer un groupe de réplication Redis (avec le mode cluster activé) pouvant contenir plusieurs groupes de nœuds (partitions), chacun avec un cluster principal et des répliques en lecture. AWS::ElastiCache::SubnetGroup Utilisez la <code>CacheSubnetGroupName</code> propriété pour spécifier le nom d'un groupe de ElastiCache sous-réseaux Amazon.	2010-05-15
Nouvelles ressources	06 octobre 20	AWS::ApiGateway::UsagePlan Utilisez la <code>AWS::ApiGateway::UsagePlan</code> ressource pour spécifier un plan d'utilisation pour le déploiement d'Amazon API Gateway APIs. AWS::CodeCommit::Repository Utilisez cette <code>AWS::CodeCommit::Repository</code> ressource pour créer un CodeCommit référentiel hébergé par Amazon Web Services.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	06 octobre 20	AWS::ApiGateway::Authorizer La propriété <code>ProviderARNs</code> permet d'utiliser des groupes d'utilisateurs Amazon Cognito comme mécanismes d'autorisation d'API Amazon API Gateway. AWS::ApiGateway::Deployment La propriété <code>StageName</code> n'est plus obligatoire. AWS::ElasticLoadBalancingV2::TargetGroup Pour la <code>GetAtt</code> fonction, utilisez l'<code>LoadBalancerArns</code> attribut pour récupérer les Amazon Resource Names (ARNs) des équilibres de charge qui acheminent le trafic vers le groupe cible. AWS::RDS::DBInstance Utilisez les propriétés <code>Domain</code> et <code>DomainIAMRoleName</code> pour utiliser l'authentification Windows lorsque les utilisateurs se connectent à l'instance de base de données RDS. AWS::EC2::SecurityGroupEgress Utilisez la <code>DestinationPrefixListId</code> propriété pour spécifier le préfixe de AWS service d'un point de terminaison Amazon VPC.	2010-05-15
Amélioration des références entre piles	06 octobre 20	Utilisez des fonctions intrinsèques pour personnaliser la valeur <code>Name</code> d'une exportation ou pour faire référence à une valeur de la fonction ImportValue .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
CloudFormation rôle de service	26 septembre 2016	Utilisez un rôle de service Gestion des identités et des accès AWS (IAM) pour les opérations de CloudFormation stack. CloudFormation utilise les informations d'identification du rôle pour passer des appels afin de cumuler des ressources en votre nom. Pour plus d'informations, consultez Rôle de service CloudFormation .	2010-05-15
Nouvelle fonctionnalité	19 septembre 2016	Vous pouvez utiliser le champ de sortie <code>Export</code> et la fonction intrinsèque <code>Fn::ImportValue</code> pour faire en sorte qu'une pile fasse référence à des sorties de ressource dans une autre pile. Pour plus d'informations, voir Sorties et procédure pas à pas : voir Sorties de ressources dans une autre CloudFormation pile . Fn::ImportValue	2010-05-15
Prise en charge de YAML	19 septembre 2016	Vous pouvez utiliser le format YAML pour créer des CloudFormation modèles. YAML vous permet aussi, par exemple, d'ajouter des commentaires à vos modèles ou d'utiliser le formulaire court pour des fonctions intrinsèques. Pour plus d'informations, consultez la section Format du CloudFormation modèle .	2010-05-15
Nouvelle fonction intrinsèque	19 septembre 2016	Utilisez la fonction <code>Fn::Sub</code> pour remplacer des variables contenues dans une chaîne d'entrée par des valeurs que vous spécifiez. Pour de plus amples informations, veuillez consulter Fn::Sub .	2010-05-15
Nouvelles ressources	19 septembre 2016	AWS::KMS::Alias Utilisez la ressource <code>AWS::KMS::Alias</code> pour créer un alias pour une AWS KMS key.	

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	19 septembre 2016	AWS::EC2::SpotFleet Pour la propriété <code>LaunchSpecifications</code> , utilisez la propriété <code>SpotPrice</code> pour spécifier un tarif d'offre pour un type d'instance spécifique. AWS::ECS::Cluster La propriété <code>ClusterName</code> permet de spécifier le nom d'un cluster Amazon Elastic Container Service. AWS::ECS::TaskDefinition Utilisez la propriété <code>TaskRoleArn</code> pour spécifier un Gestion des identités et des accès AWS rôle que les conteneurs Amazon Elastic Container Service utilisent pour passer AWS des appels en votre nom. Utilisez la propriété <code>Family</code> pour enregistrer une définition de tâche dans une famille spécifique. AWS::Elasticsearch::Domain La propriété <code>ElasticsearchVersion</code> permet de spécifier la version d'Elasticsearch à utiliser.	2010-05-15
Nouvelles ressources	11 août 2016	Les ressources d'Application Load Balancer Elastic Load Balancing suivantes permettent de répartir le trafic applicatif entrant entre plusieurs cibles, comme des instances EC2, dans plusieurs zones de disponibilité : • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressource mise à jour	11 août 2016	AWS::AutoScaling::AutoScalingGroup La propriété <code>TargetGroupARNs</code> permet d'associer le groupe Auto Scaling à un ou plusieurs groupes cibles Application Load Balancer. AWS::ECS::Service Pour la propriété <code>LoadBalancers</code>, la propriété <code>TargetGroupArn</code> permet d'associer un service Amazon Elastic Container Service à un groupe cible Application Load Balancer.	2010-05-15
Nouvelles ressources	09 août 2016	CloudFormation a ajouté les ressources suivantes : AWS::ApplicationAutoScaling::ScalableTarget et AWS::ApplicationAutoScaling::ScalingPolicy Utilisez une politique de mise à l'échelle Application Auto Scaling pour définir quand et comment mettre à l'échelle une ressource cible. AWS::CertificateManager::Certificate Fournissez un AWS Certificate Manager certificat que vous pouvez utiliser avec d'autres AWS services pour activer des connexions sécurisées.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	09 août 2016	CloudFormation a mis à jour les ressources suivantes : AWS::CloudFront::Distribution Pour la propriété <code>ViewerCertificate</code> de configuration de la distribution, vous pouvez spécifier un certificat AWS Certificate Manager . Pour la propriété <code>Origin</code> de configuration de la distribution, vous pouvez spécifier les en-têtes personnalisés et les protocoles SSL pour les origines personnalisées. AWS::EFS::FileSystem Vous pouvez spécifier le mode de performances d'un système de fichiers Amazon Elastic File System.	2010-05-15
Nouvelles ressources	20 juillet 2016	AWS IoT À utiliser pour déclarer une AWS IoT politique, un certificat X.509, une association entre une stratégie et un principal (un certificat X.509 ou autre identifiant), un AWS IoT objet, une association entre un principal et un objet, ou une règle. AWS IoT • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	20 juillet 2016	CloudFormation a mis à jour les ressources suivantes : AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User Les propriétés de nom permettent de spécifier un nom personnalisé pour les ressources Gestion des identités et des accès AWS (IAM). AWS::ApiGateway::Method Pour la propriété <code>Integration</code> , vous pouvez utiliser la propriété <code>PassthroughBehavior</code> pour spécifier à quel moment Amazon API Gateway transmet les demandes au back-end ciblé. AWS::ApiGateway::Model et AWS::ApiGateway::RestApi Vous pouvez spécifier des objets JSON pour les propriétés <code>Schema</code> et <code>Body</code>.	2010-05-15
Groupe Auto Scaling UpdatePolicy	9 juin 2016	Pour l'attribut <code>UpdatePolicy</code> , la propriété <code>AutoScalingReplacingUpdate</code> permet de préciser si un groupe Auto Scaling et les instances qu'il contient sont remplacés lorsque vous mettez à jour le groupe Auto Scaling. Lors d'un remplacement, CloudFormation conserve l'ancien groupe Auto Scaling jusqu'à ce qu'il en crée un nouveau avec succès afin de CloudFormation pouvoir revenir à l'ancien groupe Auto Scaling en cas d'échec de la mise à jour. Pour plus d'informations, consultez UpdatePolicy Attribut.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	9 juin 2016	CloudFormation a ajouté les ressources suivantes : AWS::EC2::FlowLog Crée un flux de journal Amazon Elastic Compute Cloud qui capture le trafic IP pour une interface réseau, un sous-réseau ou un VPC donné. AWS::KinesisFirehose::DeliveryStream Crée un flux de diffusion qui fournit des données de streaming en temps réel vers une destination, telle qu'Amazon Simple Storage Service, Amazon Redshift ou Amazon OpenSearch Service.	2010-05-15
Ressources mises à jour	9 juin 2016	CloudFormation a mis à jour les ressources suivantes : AWS::Kinesis::Stream La propriété Name permet de spécifier le nom d'un flux Amazon Kinesis. AWS::Lambda::Function Pour la propriété Code, vous pouvez utiliser la propriété ZipFile et le module de réponse cfn pour les environnements d'exécution nodejs4.3 . AWS::SNS::Topic CloudFormation a activé les mises à jour pour la ressource thématique Amazon Simple Notification Service.	2010-05-15
Nouvelle ressource	25 avril 2016	Utilisez la ressource AWS::EC2::Host pour attribuer un serveur physique entièrement dédié au lancement des instances EC2.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	25 avril 2016	AWS::EC2::Instance Les propriétés <code>Affinity</code> et <code>HostId</code> permettent de lancer des instances sur un hôte dédié Amazon Elastic Compute Cloud. AWS::ECS::Service Utilisez la propriété <code>DeploymentConfiguration</code> pour configurer le nombre de tâches pouvant être exécutées lors d'un déploiement. AWS::ECS::TaskDefinition CloudFormation ajout de la prise en charge des propriétés supplémentaires de définition de conteneur Amazon Elastic Container Service. AWS::GameLift::Fleet Utilisez les <code>MinSize</code> propriétés <code>MaxSize</code> et pour spécifier le nombre maximum et minimum d'instances EC2 autorisées dans votre parc Amazon GameLift Servers. AWS::Lambda::Function Utilisez la <code>FunctionName</code> propriété pour définir le nom de votre AWS Lambda fonction. Vous pouvez également utiliser Python 2.7 pour spécifier une fonction en ligne.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	18 avril 2016	Amazon API Gateway Utilisez les ressources Amazon API Gateway pour publier, gérer et surveiller les API à n'importe quel niveau. Vous pouvez créer des API que les clients peuvent appeler pour accéder aux services back-end, telles que les applications qui exécutent des instances EC2 ou un code exécuté dans AWS Lambda. • AWS::ApiGateway::Account • AWS::ApiGateway::ApiKey • AWS::ApiGateway::Authorizer • AWS::ApiGateway::BasePathMapping • AWS::ApiGateway::ClientCertificate • AWS::ApiGateway::Deployment • AWS::ApiGateway::Method • AWS::ApiGateway::Model • AWS::ApiGateway::Resource • AWS::ApiGateway::RestApi • AWS::ApiGateway::Stage AWS::Events::Rule Créez une règle Amazon CloudWatch Events qui surveille les modifications apportées aux AWS ressources de votre compte (événements). Si un événement entrant répond aux conditions que vous avez décrites dans la règle, Amazon CloudWatch Events envoie des messages aux cibles que vous avez spécifiées et les active, telles que AWS Lambda les fonctions ou les rubriques Amazon Simple Notification Service.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::WAF::SizeConstraintSet et AWS::WAF::XssMatchSet Utilisez les deux AWS WAF règles pour vérifier la taille d'une requête Web ou pour empêcher les attaques par script intersite.	
Nouvelles ressources	31 mars 2016	Utilisez la AWS::Lambda::Alias ressource pour créer des alias pour vos AWS Lambda fonctions et la AWS::Lambda::Version ressource pour créer des versions de vos fonctions.	2010-05-15
Ressources mises à jour	31 mars 2016	CloudFormation a mis à jour les ressources suivantes : AWS::EMR::Cluster et AWS::EMR::InstanceGroupConfig La propriété <code>EbsConfiguration</code> vous permet de configurer des volumes de stockage Amazon Elastic Block Store pour vos clusters ou groupes d'instances Amazon EMR. AWS::Lambda::Function Utilisez la propriété <code>VpcConfig</code> pour permettre aux fonctions AWS Lambda d'accéder aux ressources d'un VPC. AWS::S3::Bucket Pour les règles de cycle de vie Amazon Simple Storage Service, vous pouvez spécifier plusieurs règles de transition qui déterminent à quel moment les objets passent à une classe de stockage donnée.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Jeux de modifications	29 mars 2016	Avant de mettre à jour les piles, utilisez des jeux de modifications pour voir l'impact possible des modifications sur les ressources exécutées. Pour plus d'informations, consultez Mise à jour de piles à l'aide de jeux de modifications .	2010-05-15
Nouvelles ressources	15 mars 2016	Utilisez les AWS::GameLift::Fleet ressources AWS::GameLift::Alias AWS::GameLift::Build , et pour déployer des serveurs de jeux multijoueurs dans AWS.	2010-05-15
Nouvelles ressources	26 février 201	CloudFormation a ajouté les ressources suivantes : AWS::ECR::Repository Créez des référentiels Amazon Elastic Container Registry dans lesquels les utilisateurs peuvent importer et télécharger des images Docker. AWS::EC2::NatGateway Utilisez la passerelle de traduction d'adresse réseau (NAT) pour permettre aux instances EC2 d'un sous-réseau privé de se connecter à Internet. AWS::Elasticsearch::Domain Créez des domaines Amazon OpenSearch Service qui exécutent les anciens clusters Elasticsearch OSS. AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step Utiliser les ressources Amazon EMR pour vous aider à analyser et à traiter de grandes quantités de données. Vous pouvez créer des clusters, puis exécuter des tâches annexes.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Ressources mises à jour	26 février 201	CloudFormation a mis à jour les ressources suivantes : AWS::CloudTrail::Trail Utilisez la <code>IsMultiRegionTrail</code> propriété pour spécifier s'il faut créer un AWS CloudTrail sentier dans la région dans laquelle vous créez une pile ou dans toutes les régions. AWS::Config::ConfigurationRecorder Pour le groupe d'enregistrement, utilisez la propriété <code>IncludeGlobalResourceTypes</code> afin d'enregistrer tous les types de ressources globales. AWS::RDS::DBCluster Utilisez les propriétés <code>KmsKeyId</code> et <code>StorageEncrypted</code> pour chiffrer les instances de base de données du cluster.	2010-05-15
Conservation des ressources	26 février 201	Pour les piles présentant l'état <code>DELETE_FAILED</code> , le paramètre <code>RetainResources</code> permet de conserver les ressources qui ne peuvent pas être supprimées par CloudFormation . Pour plus d'informations, consultez Échec de la suppression de la pile.	2010-05-15
Mise à jour des balises de pile	26 février 201	Vous pouvez ajouter, modifier ou supprimer les balises de la pile lorsque vous mettez à jour cette dernière. Pour plus d'informations, consultez CloudFormation Stacks Updates.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Restauration de l'état de fonctionnement de la pile en cas d'échec de la mise à jour	25 janvier 2016	Pour une pile dont l'état indique UPDATE_ROLLBACK_FAILED , vous pouvez continuer à restaurer la mise à jour afin de rétablir l'état de fonctionnement de la pile. De cette façon, vous pouvez rétablir les paramètres d'origine de la pile et essayer de la mettre à jour à nouveau. Pour plus d'informations, consultez Poursuite de la restauration d'une mise à jour .	2010-05-15
Nouveaux exemples de modèles disponibles pour la région Asie-Pacifique (Séoul).	7 janvier 2016	La collection d' CloudFormation exemples de modèles suivante concerne la région ap-northeast-2 : - Exemples de solutions - Frameworks applicatifs - Services Pour plus d'informations, consultez la section Utilisation des CloudFormation modèles .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	28 décembre 2015	CloudFormation a ajouté les ressources suivantes : AWS::DirectoryService::MicrosoftAD Utilisez la ressource Microsoft Active Directory pour créer un répertoire Microsoft Active Directory dans AWS. AWS::Logs::Destination et AWS::Logs::LogStream Utilisez les ressources Amazon CloudWatch Logs pour créer une destination pour le traitement en temps réel des données de journal ou pour créer des flux de journaux, respectivement. AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet et AWS::WAF::WebACL Utilisez les AWS WAF ressources pour contrôler et surveiller les requêtes Web relatives à votre contenu.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mises à jour de ressource	28 décembre 2015	CloudFormation a mis à jour les ressources suivantes : AWS::CloudFront::Distribution Pour la configuration de la distribution, utilisez la WebACLIId propriété pour associer une liste de contrôle d'accès AWS WAF Web (ACL) à une CloudFront distribution Amazon. Pour le comportement de cache et le comportement de cache par défaut, vous pouvez spécifier une durée de vie (TTL) par défaut et minimale. AWS::DynamoDB::Table Vous pouvez créer, mettre à jour ou supprimer un index secondaire global sans avoir à remplacer votre table Amazon DynamoDB. AWS::S3::Bucket Utilisez la propriété ReplicationConfiguration pour spécifier les objets à répliquer et l'endroit où ils sont stockés. Les propriétés de la propriété NotificationConfiguration permettent de spécifier des filtres de sorte qu'Amazon Simple Storage Service envoie des notifications pour les objets que vous spécifiez.	2010-05-15
Regroupement et tri des paramètres	3 décembre 2015	Utilisez la clé de AWS::CloudFormation::Interfacedata pour regrouper et trier les paramètres dans la CloudFormation console lorsque les utilisateurs créent ou mettent à jour une pile avec votre modèle.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Attribut UpdatePolicy	3 décembre 2015	Pour un attribut de politique de mise à jour Auto Scaling, utilisez la propriété <code>MinSuccessfulInstancesPercent</code> pour spécifier le pourcentage d'instances qui doivent envoyer un signal de réussite pour qu'une mise à jour aboutisse.	2010-05-15
Nouvelles ressources	3 décembre 2015	CloudFormation a ajouté les ressources suivantes : AWS::CodePipeline::Pipeline et AWS::CodePipeline::CustomActionType Utilisez les CodePipeline ressources pour créer un pipeline qui décrit le processus de publication des modifications logicielles. AWS::Config::ConfigurationRecorder, AWS::Config::DeliveryChannel et AWS::Config::ConfigRule Utilisez les AWS Config ressources pour surveiller les modifications de configuration apportées à des AWS ressources spécifiques. AWS::KMS::Key Utilisez la ressource AWS Key Management Service (AWS KMS) pour créer des clés gérées par le client AWS KMS que les utilisateurs peuvent utiliser pour chiffrer de petites quantités de données. AWS::SSM::Document Utilisez la ressource Amazon EC2 Systems Manager pour créer un document qui spécifie les configurations dans les instances.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mise à jour des ressources	3 décembre 2015	CloudFormation a mis à jour les ressources suivantes : AWS::AutoScaling::LaunchConfiguration Spécifiez si les volumes EBS sont chiffrés. AWS::AutoScaling::ScalingPolicy Vous pouvez utiliser deux types de politiques différents (simple et échelonnement par étapes) pour spécifier la manière dont un groupe Auto Scaling évolue en cas de violation d'une alarme Amazon CloudWatch (CloudWatch). AWS::CloudTrail::Trail Utilisez les CloudWatch propriétés pour envoyer des journaux à un groupe de CloudWatch journaux. Vous pouvez ajouter des balises à un parcours et spécifier celles AWS KMS key que vous souhaitez utiliser pour chiffrer les journaux. AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentConfig et AWS::CodeDeploy::DeploymentGroup Utilisez les <code>DeploymentGroupName</code> propriétés <code>ApplicationName</code> <code>DeploymentConfigName</code> , et pour spécifier des noms personnalisés pour les CodeDeploy ressources. AWS::DynamoDB::Table La propriété <code>StreamSpecification</code> permet de spécifier les paramètres de capture des modifications apportées aux éléments stockés dans une table Amazon DynamoDB (DynamoDB).	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::EC2::Instance La propriété <code>SsmAssociations</code> permet d'associer un document Amazon EC2 Systems Manager à une instance. AWS::EC2::SpotFleet Utilisez la propriété <code>AllocationStrategy</code> pour spécifier la répartition de la capacité cible entre les groupes d'instances Spot. Utilisez la propriété <code>ExcessCapacityTerminationPolicy</code> pour spécifier la façon dont les instances sont mises hors service si la capacité cible est inférieure à la taille du parc d'instances Spot. AWS::Redshift::Cluster Utilisez la propriété AWS KMS key pour spécifier une <code>KmsKeyId</code> pour chiffrer les données dans un cluster Amazon Redshift. AWS::WorkSpaces::Workspace Utilisez les propriétés de chiffrement pour chiffrer les données stockées sur des volumes.	
Mise à jour de ressource	4 novembre 2010	Pour la AWS::EC2::Volume ressource, utilisez la propriété <code>AutoEnableIO</code> pour reprendre automatiquement les I/O opérations si les données d'un volume deviennent incohérentes.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	1er octobre 2010	CloudFormation a ajouté les ressources suivantes : AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentGroup et AWS::CodeDeploy::DeploymentConfig Utilisez les CodeDeploy ressources pour créer et appliquer des déploiements sur des instances EC2 ou sur site. AWS::DirectoryService::SimpleAD Utilisez la ressource Simple Active Directory pour créer un annuaire Directory Service Simple AD, qui est compatible avec Microsoft Active Directory. AWS::EC2::PlacementGroup Utilisez un groupe de placement pour créer un cluster d'instances dans un réseau à faible latence. AWS::EC2::SpotFleet Utilisez un parc d'instances Spot pour lancer une collection d'instances Spot qui exécutent des tâches qui peuvent être interrompues. AWS::Lambda::EventSourceMapping Utilisez la ressource de mappage des sources d'événements pour spécifier un flux en tant que source d'événements pour une AWS Lambda fonction (Lambda). AWS::Lambda::Permission Une autorisation Lambda permet d'ajouter une instruction à la politique d'une fonction Lambda.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		AWS::Logs::SubscriptionFilter Utilisez le filtre de l'abonnement pour définir quels événements de journal sont fournis à votre flux Kinesis. AWS::RDS::DBCluster et AWS::RDS::DBClusterParameterGroup Utilisez le cluster et les ressources du groupe de paramètres du cluster pour créer un cluster de base de données Amazon Aurora. AWS::WorkSpaces::Workspace Utilisez-le WorkSpaces pour créer des expériences de bureau basées sur le cloud.	

Modifier	Date de parution	Description	Version de l'API
Mises à jour de ressource	1er octobre 2010	CloudFormation a mis à jour les ressources suivantes : AWS::ElastiCache::ReplicationGroup Utilisez la fonction intrinsèque <code>Fn::GetAtt</code> pour obtenir une liste des ports et adresses de réplica en lecture seule. AWS::OpsWorks::Stack Utilisez la <code>AgentVersion</code> propriété pour spécifier un OpsWorks agent en particulier. AWS::OpsWorks::App Utilisez la <code>Environment</code> propriété pour spécifier les variables d'environnement d'une OpsWorks application. AWS::S3::Bucket Pour cette NotificationConfiguration propriété, vous pouvez configurer les paramètres de notification pour les fonctions Lambda et les files d'attente Amazon Simple Queue Service (Amazon SQS).	2010-05-15
Clés de condition IAM	1er octobre 2010	Pour les politiques Gestion des identités et des accès AWS (IAM), utilisez des clés de condition CloudFormation spécifiques pour spécifier à quel moment une politique IAM entre en vigueur. Pour plus d'informations, consultez Controlling Access with AWS Identity and Access Management.	2010-05-15
CloudFormation Designer	1er octobre 2010	Utilisez CloudFormation Designer pour créer et modifier des modèles à l'aide d'une drag-and-drop interface.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle ressource	24 août 2015	Utilisez la AWS::EC2::VPCEndpoint ressource pour établir une connexion privée entre votre VPC et un autre AWS service.	2010-05-15
Mises à jour de ressource	24 août 2015	CloudFormation a mis à jour les ressources suivantes : AWS::ElasticBeanstalk::Environment Utilisez la Tags propriété pour spécifier des balises (paires clé-valeur) pour un environnement AWS Elastic Beanstalk (Elastic Beanstalk). AWS::Lambda::Function Pour la propriété Code, utilisez la propriété ZipFile pour écrire le code source de la fonction Lambda directement dans un modèle. Actuellement, vous pouvez uniquement utiliser la propriété ZipFile pour les environnements d'exécution nodejs. Vous pouvez continuer à renvoyer vers un fichier d'un compartiment S3 pour tous les environnements d'exécution, tels que java8 et nodejs. AWS::OpsWorks::Instance La propriété EbsOptimized permet d'indiquer si une instance est optimisée pour les E/S Amazon Elastic Block Store (Amazon EBS). AWS::RDS::DBInstance Pour la propriété SourceDBInstanceId entifier , vous pouvez spécifier une instance de base de données dans une autre région pour créer un réplica en lecture entre régions.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
URL de modèle Amazon S3	24 août 2015	Pour les compartiments dans lesquels la gestion des versions est activée, vous pouvez spécifier un ID de version dans une URL de modèle Amazon S3 lorsque vous créez ou mettez à jour une pile, par exemple <code>https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTdW</code> .	2010-05-15
Nouvelle ressource	3 août 2015	La ressource AWS::EFS::FileSystem permet de créer un système de fichiers Amazon Elastic File System (Amazon EFS) et la ressource AWS::EFS::MountTarget de créer un point de montage pour un système de fichiers.	2010-05-15
Modification des exigences d'autorisation	11 juin 2015	Lorsque vous créez ou mettez à jour une ressource AWS::RDS::DBInstance , vous devez désormais également avoir l'autorisation d'appeler l'action <code>ec2:DescribeAccountAttributes</code> .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	11 juin 2015	CloudFormation a ajouté les ressources suivantes : AWS::DataPipeline::Pipeline Utilisez les pipelines de données pour automatiser le mouvement et la transformation de données. Ressources Amazon Elastic Container Service Utilisez les ressources AWS::ECS::Service, AWS::ECS::Cluster et AWS::ECS::TaskDefinition pour créer des conteneurs Docker dans un cluster d'instances EC2. AWS::ElastiCache::ReplicationGroup Utilisez des groupes de réplication pour créer une collection de nœuds avec un cluster principal en lecture/écriture et un maximum de cinq clusters secondaires en lecture seule. AWS::IAM::ManagedPolicy Utilisez des politiques gérées pour créer des politiques dans votre AWS compte que vous pouvez utiliser pour appliquer des autorisations aux utilisateurs, aux groupes et aux rôles IAM. AWS::Lambda::Function Utilisez les fonctions Lambda pour exécuter du code en réponse à des événements. AWS::RDS::OptionGroup Utilisez des groupes d'options pour vous aider à créer et à gérer des bases de données Amazon Relational Database Service (Amazon RDS).	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mises à jour de ressource	11 juin 2015	CloudFormation a mis à jour les ressources suivantes : AWS::EC2::Subnet Utilisez la propriété <code>MapPublicIpOnLaunch</code> pour attribuer automatiquement une adresse IP publique aux instances d'un sous-réseau. AWS::ElastiCache::CacheCluster Utilisez la propriété <code>SnapshotName</code> pour restaurer les données d'instantané dans un nouveau cluster de cache Redis. AWS::IAM::User Pour la propriété <code>LoginProfile</code> , la propriété <code>PasswordResetRequired</code> permet de contraindre les utilisateurs à définir un nouveau mot de passe lorsqu'ils se connectent à AWS Management Console. AWS::OpsWorks::Layer Utilisez cette <code>LifecycleEventConfiguration</code> propriété pour configurer les événements du cycle de vie d'une OpsWorks couche. AWS::S3::Bucket Pour la propriété <code>LifecycleConfiguration</code> , utilisez les propriétés <code>NoncurrentVersionExpirationInDays</code> et <code>NoncurrentVersionTransition</code> pour spécifier les règles du cycle de vie pour les versions d'objets non actuelles.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouveaux types de paramètres	19 mai 2015	Chaque fois que vous utilisez la CloudFormation console pour créer ou mettre à jour une pile, vous pouvez rechercher des valeurs de type de paramètre AWS spécifiques par ID, nom ou valeur de balise Name. CloudFormation a également ajouté la prise en charge des types de paramètres AWS spécifiques suivants. Pour plus d'informations, consultez Parameters. • <code>AWS::EC2::AvailabilityZone::Name</code> • <code>List<AWS::EC2::AvailabilityZone ::Name></code> • <code>AWS::EC2::Instance::Id</code> • <code>List<AWS::EC2::Instance ::Id></code> • <code>AWS::EC2::Image::Id</code> • <code>List<AWS::EC2::Image::Id ></code> • <code>AWS::EC2::SecurityGroup::GroupName</code> • <code>List<AWS::EC2::SecurityGroup::Gr oupName ></code> • <code>AWS::EC2::Volume::Id</code> • <code>List<AWS::EC2::Volume::Id ></code> • <code>AWS::Route53::HostedZone::Id</code> • <code>List<AWS::Route53::HostedZone ::Id></code>	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles ressources	16 avril 2015	CloudFormation a ajouté les ressources suivantes : AWS::AutoScaling::LifecycleHook Utilisez les hooks de cycle de vie Auto Scaling pour contrôler l'état d'une instance après son lancement ou son arrêt. AWS::RDS::EventSubscription Utilisez des abonnements aux événements pour recevoir des notifications concernant les événements Amazon RDS.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mises à jour de ressource	16 avril 2015	CloudFormation a mis à jour les ressources suivantes : AWS::AutoScaling::AutoScalingGroup Utilisez la propriété <code>NotificationConfigurations</code> pour spécifier plusieurs notifications. AWS::AutoScaling::LaunchConfiguration Utilisez la propriété <code>PlacementTenancy</code> pour spécifier la location d'instances. Utilisez les <code>ClassicLinkVPCSecurityGroups</code> propriétés <code>ClassicLinkVPCId</code> et pour lier les instances EC2-Classique à un VPC ClassicLink activé. AWS::AutoScaling::ScalingPolicy Utilisez la propriété <code>MinAdjustmentStep</code> pour spécifier le nombre minimum d'instances qui sont ajoutées ou supprimées lors d'un événement de mise à l'échelle. AWS::CloudFront::Distribution Pour les certificats de visionneuse, utilisez la propriété <code>MinimumProtocolVersion</code> pour spécifier une version minimale de protocole. Pour les comportements du cache, utilisez la <code>CachedMethods</code> propriété pour spécifier les méthodes pour lesquelles Amazon CloudFront (CloudFront) met en cache les réponses. Pour les origines, utilisez le <code>OriginPath</code> pour spécifier un chemin à CloudFront utiliser pour demander du contenu.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		<u>AWS::ElastiCache::CacheCluster</u> Pour les clusters de cache Memcached, utilisez les <code>PreferredAvailabilityZones</code> propriétés <code>AZMode</code> et pour spécifier des nœuds dans plusieurs zones de disponibilité (AZs). <u>AWS::EC2::Volume</u> Utilisez la propriété <code>KmsKeyId</code> pour spécifier une clé gérée par un client pour les volumes chiffrés. <u>AWS::OpsWorks::Instance</u> Utilisez la propriété <code>TimeBasedAutoScaling</code> pour dimensionner automatiquement les instances en fonction d'une planification que vous spécifiez. <u>AWS::OpsWorks::Layer</u> Utilisez la propriété <code>LoadBasedAutoScaling</code> pour spécifier des politiques de mise à l'échelle basées sur la charge. Pour les configurations de volume, utilisez les Iops propriétés <code>VolumeType</code> et pour spécifier respectivement un type de volume et le nombre d' I/O opérations par seconde. <u>AWS::RDS::DBInstance</u> Utilisez la propriété <code>CharacterSetName</code> pour spécifier un jeu de caractères pour les moteurs de base de données pris en charge. Utilisez la propriété <code>StorageEncrypted</code> pour indiquer si les instances de base de données sont chiffrées, et la propriété <code>KmsKeyId</code> pour spécifier une clé gérée par un client pour les instances de base de données chiffrées.	

Modifier	Date de parution	Description	Version de l'API
		AWS::Route53::HealthCheck Utilisez la propriété <code>HealthCheckTags</code> pour associer les balises à des vérifications de l'état. AWS::Route53::HostedZone Utilisez la propriété <code>VPCs</code> pour créer des zones hébergées privées. Utilisez la propriété <code>HostedZoneTags</code> pour associer les balises à des zones hébergées.	
Nouvelle section de modèle	16 avril 2015	Ajoutez la section Metadata à vos modèles afin d'inclure des objets JSON arbitraires qui décrivent ces modèles, tels que des détails de conception ou de mise en œuvre.	2010-05-15
Mise à jour de ressource	8 avril 2015	Pour la AWS::CloudFormation::CustomResource ressource, vous pouvez spécifier la fonction Lambda Amazon Resource Names (ARNs) dans la <code>ServiceToken</code> propriété.	2010-05-15
Mise à jour Amazon RDS	24 décembre 2014	CloudFormation a ajouté deux nouvelles propriétés pour les instances de base de données RDS. Vous pouvez associer un groupe d'options à une instance DB et spécifier le type de stockage d'instance DB. Pour de plus amples informations, veuillez consulter AWS::RDS::DBInstance .	2010-05-15
Mise à jour Elastic Load Balancing	24 décembre 2014	Vous pouvez utiliser la propriété <code>ConnectionSettings</code> pour spécifier combien de temps les connexions peuvent rester inactives. Pour de plus amples informations, veuillez consulter AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mise à jour de Route 53	6 novembre 2010	Vous pouvez maintenant allouer et gérer des zones hébergées Route 53, des surveillances de l'état , des jeux d'enregistrements de basculement et des jeux d'enregistrements de géolocalisation .	2010-05-15
Amélioration de mise à jour propagée Auto Scaling	6 novembre 2010	Lors d'une mise à jour, vous pouvez utiliser l' <code>WaitOnResourceSignals</code> indicateur pour demander d'CloudFormation attendre que les instances signalent le succès. De cette façon, CloudFormation ne met pas à jour le lot d'instances suivant tant que le lot actuel n'est pas prêt. Pour plus d'informations, consultez Attribut UpdatePolicy .	2010-05-15
Nouveau VPC Fn : attributs GetAtt	6 novembre 2010	Avec un ID de VPC, vous pouvez récupérer le groupe de sécurité par défaut et l'ACL réseau de ce VPC. Pour de plus amples informations, veuillez consulter Fn::GetAtt .	2010-05-15
Nouveaux AWS types de paramètres spécifiques	6 novembre 2010	Vous pouvez spécifier des types de paramètres AWS spécifiques dans vos CloudFormation modèles. Dans la CloudFormation console, ces types de paramètres fournissent une liste déroulante de valeurs valides. Avec l'API AWS CLI, CloudFormation vous pouvez rapidement valider les valeurs de ces types de paramètres avant de créer ou de mettre à jour une pile. Pour plus d'informations, consultez Parameters .	2010-05-15
CreationPolicy attribut	6 novembre 2010	Avec CreationPolicy cet attribut, vous pouvez demander d'attendre que CloudFormation les applications soient prêtes sur les instances EC2 avant de procéder à la création de la pile. Vous pouvez utiliser une politique de création au lieu d'une condition d'attente et d'un descripteur de condition d'attente. Pour plus d'informations, consultez Attribut CreationPolicy .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Amazon CloudFront a transféré des valeurs	29 septembre 2014	Pour les comportements de cache, vous pouvez transférer les en-têtes à l'origine. Consultez Forwarded Values .	2010-05-15
AWS OpsWorks update	29 septembre 2014	Pour Chef 11.10, vous pouvez utiliser la propriété <code>ChefConfiguration</code> afin d'activer Berkshelf. Vous pouvez également utiliser les groupes de sécurité AWS OpsWorks intégrés avec vos AWS OpsWorks piles. Pour de plus amples informations, veuillez consulter AWS::OpsWorks::Stack .	2010-05-15
Prise en charge du balisage Elastic Load Balancing	29 septembre 2014	AWS CloudFormation étiquette les équilibreurs de charge Elastic Load Balancing dotés de balises au niveau de la pile. Vous pouvez également ajouter vos propres balises à un équilibreur de charge. Consultez AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
Mise à jour de la politique de rubrique Amazon Simple Notification Service	29 septembre 2014	Vous pouvez maintenant mettre à jour les politiques de rubrique Amazon SNS. Pour de plus amples informations, veuillez consulter AWS::SNS::TopicPolicy .	2010-05-15
Mise à jour d'instance DB RDS	5 septembre 2014	Vous pouvez spécifier si une instance de base de données est accessible depuis Internet en utilisant la propriété <code>PubliclyAccessible</code> dans la ressource AWS::RDS::DBInstance .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
UpdatePolicy mise à jour des attributs	05 septembre 2014	Vous pouvez spécifier une politique de mise à jour pour un groupe Auto Scaling auquel une action planifiée est associée. Pour plus d'informations, consultez Attribut UpdatePolicy .	2010-05-15
CloudWatch Assistance Amazon	10 juillet 2014	Vous pouvez l'utiliser CloudFormation pour provisionner et gérer les groupes de CloudWatch journaux et les filtres métriques Amazon CloudWatch Logs (Logs). Pour plus d'informations, consultez AWS::Logs::LogGroup ou AWS::Logs::MetricFilter .	2010-05-15
Mise à jour CloudFront de la configuration de la distribution Amazon	17 juin 2014	Vous pouvez spécifier des propriétés CloudFront de configuration de distribution supplémentaires : • Les réponses d'erreur personnalisées permettent de définir des messages d'erreur personnalisés pour les codes d'état HTTP 4xx et 5xx. • La catégorie de tarifs définit le prix maximum que vous souhaitez payer pour le service CloudFront . • Les restrictions définissent qui peut afficher votre contenu. • Le certificat de visionneuse spécifie le certificat à utiliser lorsque les visionneuses utilisent le protocole HTTPS. • Pour les comportements de cache, vous pouvez spécifier les méthodes HTTP autorisées et indiquer si les cookies doivent être transférés. Pour de plus amples informations, veuillez consulter AWS::CloudFront::Distribution.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mise à jour d'instance EC2	17 juin 2014	Vous pouvez spécifier si une instance s'arrête ou devient hors service lorsque vous appelez la commande de fermeture du système d'exploitation de l'instance. Pour de plus amples informations, veuillez consulter AWS::EC2::Instance .	2010-05-15
Mise à jour de volume EBS	17 juin 2014	Vous pouvez utiliser les volumes EBS chiffrés avec les types d'instance pris en charge. Pour de plus amples informations, veuillez consulter AWS::EC2::Volume .	2010-05-15
Nouvelle connexion d'appairage de Amazon VPC	17 juin 2014	Vous pouvez l'utiliser CloudFormation pour créer une connexion d'appairage Amazon Virtual Private Cloud (Amazon VPC), qui établit une connexion réseau entre deux VPCs. Pour de plus amples informations, veuillez consulter AWS::EC2::VPCPeeringConnection .	2010-05-15
Mise à jour de groupe Amazon EC2 Auto Scaling	17 juin 2014	Vous pouvez spécifier un groupe de placement de cluster existant dans lequel vous lancerez les instances d'un groupe Amazon EC2 Auto Scaling. Pour de plus amples informations, veuillez consulter AWS::AutoScaling::AutoScalingGroup .	2010-05-15
AWS CloudTrail soutien	17 juin 2014	CloudFormation supports AWS CloudTrail, qui peuvent capturer les appels d'API effectués depuis votre AWS compte et publier les journaux à l'emplacement que vous désignez. Pour de plus amples informations, veuillez consulter AWS::CloudTrail::Trail .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Améliorations de la mise à jour des piles	12 mai 2014	CloudFormation prend en charge des fonctionnalités supplémentaires pour la mise à jour des piles : • Vous pouvez mettre à jour les paramètres de la CloudFormation pile sans soumettre à nouveau le modèle de la pile. • Vous pouvez ajouter ou supprimer des rubriques de notification Amazon SNS pour une pile CloudFormation. Pour plus d'informations, consultez CloudFormation Stacks Updates.	2010-05-15
Prise en charge d'Amazon Kinesis	6 mai 2014	Vous pouvez l'utiliser CloudFormation pour créer des flux Amazon Kinesis qui capturent et transportent des enregistrements de données à partir de sources de données. Pour de plus amples informations, veuillez consulter AWS::Kinesis::Stream.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelles propriétés de compartiment S3	5 mai 2014	CloudFormation prend en charge des propriétés de compartiment S3 supplémentaires : • Le partage des ressources cross-origin (CORS) définit le partage de ressources cross-origin des objets dans un compartiment. • La gestion du cycle de vie définit la façon dont Amazon S3 gère les objets durant leur cycle de vie. • La politique de journalisation de l'accès enregistre les informations relatives aux requêtes soumises à votre compartiment. • Les notifications permettent de définir les événements à signaler et la rubrique Amazon SNS dans laquelle envoyer des messages. • La gestion des versions permet de stocker plusieurs variantes de tous les objets dans un compartiment. • Les règles de routage et de redirection régissent le comportement de redirection des requêtes soumises au point de terminaison de site web d'un compartiment. Pour de plus amples informations, veuillez consulter AWS::S3::Bucket.	2010-05-15
Prise en charge d'Amazon EC2 Auto Scaling	5 mai 2014	CloudFormation prend en charge la collecte de métriques pour un groupe Auto Scaling. Pour de plus amples informations, veuillez consulter AWS::AutoScaling::AutoScalingGroup.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Mise à jour Fn::If	5 mai 2014	Vous pouvez utiliser la fonction intrinsèque Fn::If dans la section de sortie d'un modèle. Pour plus d'informations, consultez Fonctions de condition .	2010-05-15
Journalisation de l'API avec AWS CloudTrail	2 avril 2014	Vous pouvez utiliser AWS CloudTrail (CloudTrail) pour enregistrer les CloudFormation demandes. CloudTrail Vous pouvez ainsi obtenir un historique des appels d' CloudFormation API pour votre compte. Pour plus d'informations, consultez la section Journalisation des appels d' CloudFormation API avec AWS CloudTrail .	2010-05-15
Mise à jour Elastic Load Balancing	20 mars 2014	Vous pouvez spécifier une politique de journalisation de l'accès afin de capturer des informations sur les requêtes soumises à l'équilibreur de charge. Vous pouvez également définir une politique de drainage de la connexion qui décrit comment gérer les requêtes en cours lorsque les instances sont désinscrites ou qu'elles deviennent défectueuses. Pour de plus amples informations, veuillez consulter AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
OpsWorks soutien	3 mars 2014	Vous pouvez l'utiliser CloudFormation pour approvisionner et gérer des OpsWorks piles. Pour plus d'informations, consultez notre section AWS::OpsWorks::StackAWS OpsWorks Extraits de modèle .	2010-05-15
Augmentation de la limite de taille des modèles Amazon S3	18 février 201	Vous pouvez spécifier des tailles de modèle pouvant atteindre 460 800 octets dans Amazon S3.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Prise en charge d'Amazon Redshift	10 février 201	Vous pouvez l'utiliser CloudFormation pour provisionner et gérer des clusters Amazon Redshift. Pour plus d'informations, consultez la section Extraits de code de modèle Amazon Redshift (français non garanti) ou AWS::Redshift::Cluster (français non garanti).	2010-05-15
Compartiments S3 et mise à jour des politiques de compartiment	10 février 201	Vous pouvez mettre à jour certaines propriétés du compartiment S3 et des ressources de politique de compartiment. Pour plus d'informations, consultez AWS::S3::Bucket ou AWS::S3::BucketPolicy .	2010-05-15
Environnements Elastic Beanstalk et mise à jour des versions d'application	10 février 201	Vous pouvez mettre à jour les configurations des environnements Elastic Beanstalk et les versions d'applications. Pour plus d'informations, consultez AWS::ElasticBeanstalk::Environment , AWS::ElasticBeanstalk::ConfigurationTemplate ou AWS::ElasticBeanstalk::ApplicationVersion .	2010-05-15
Mise à jour Amazon SQS	29 janvier 201	Vous pouvez spécifier une file d'attente de lettre morte pour une file d'attente Amazon SQS. Pour de plus amples informations, veuillez consulter AWS::SQS::Queue .	2010-05-15
Actions planifiées Auto Scaling	27 janvier 201	Vous pouvez ajuster le nombre d'instances EC2 dans un groupe Auto Scaling sur la base d'une planification. Grâce à une planification, vous pouvez mettre à l'échelle les applications en réponse à des modifications de charge prévisibles. Pour de plus amples informations, veuillez consulter AWS::AutoScaling::ScheduledAction .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Index secondaires DynamoDB	27 janvier 2014	Vous pouvez créer des index secondaires locaux et globaux pour les bases de données DynamoDB. Les index secondaires vous permettent d'accéder efficacement aux données avec des attributs autres que la clé primaire. Pour de plus amples informations, veuillez consulter AWS::DynamoDB::Table .	2010-05-15
Mise à jour Auto Scaling	2 janvier 2014	Vous pouvez spécifier un ID d'instance pour un groupe Auto Scaling ou une configuration de lancement. Vous pouvez aussi spécifier des propriétés Auto Scaling supplémentaires de périphérique de stockage en mode bloc. Pour plus d'informations, consultez AWS::AutoScaling::AutoScalingGroup ou AWS::AutoScaling::LaunchConfiguration .	2010-05-15
Mise à jour Amazon SQS	2 janvier 2014	Vous pouvez mettre à jour les files d'attente SQS et spécifier des propriétés supplémentaires. Pour de plus amples informations, veuillez consulter AWS::SQS::Queue .	2010-05-15
Augmentation des limites	2 janvier 2014	Vous pouvez spécifier jusqu'à 60 paramètres et 60 sorties dans vos modèles CloudFormation.	2010-05-15
Nouvelle console	19 décembre 2013	La nouvelle CloudFormation console ajoute des fonctionnalités telles que l'actualisation automatique des événements de pile et l'ordre alphabétique des paramètres de pile.	2010-05-15
Équilibrage de charge entre zones	19 décembre 2013	Grâce à l'équilibrage de charge entre zones, vous pouvez acheminer le trafic vers les instances principales dans toutes les zones de disponibilité (AZs). Pour de plus amples informations, veuillez consulter AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
AWS Elastic Beanstalk niveaux d'environnement	19 décembre 2013	Vous pouvez spécifier si des ressources AWS Elastic Beanstalk sont allouées pour prendre en charge un serveur Web ou pour gérer les tâches de traitement en arrière-plan. Pour de plus amples informations, veuillez consulter AWS::ElasticBeanstalk::Environment .	2010-05-15
Noms des ressources	19 décembre 2013	Vous pouvez attribuer des noms (physiques IDs) aux ressources suivantes : • ElastiCache clusters • Equilibreurs de charge Elastic Load Balancing • Instances de base de données RDS Pour plus d'informations, consultez Type de nom.	2010-05-15
Prise en charge de VPN	22 novembre	Vous pouvez permettre à une passerelle réseau privé virtuel de propager des routages aux tables de routage d'un VPC. Pour de plus amples informations, veuillez consulter AWS::EC2::VPNGatewayRoutePropagation .	2010-05-15
Création de ressources et attribution de propriétés de manière conditionnelle	8 novembre 2	Les paramètres d'entrée vous permettent de contrôler la création et les paramètres des ressources données d'une pile en définissant les conditions dans les modèles CloudFormation. Par exemple, vous pouvez utiliser des conditions pour créer les ressources d'une pile pour un environnement de production. Avec ce même modèle, vous pouvez créer les ressources d'une pile similaire avec une capacité plus faible pour un environnement de test. Pour plus d'informations, consultez Fonctions de condition .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Empêcher les mises à jour accidentelles des ressources d'une pile	8 novembre 2010	Vous pouvez empêcher les mises à jour qui peuvent entraîner des changements involontaires au niveau des ressources de la pile. Par exemple, si vous avez une pile avec une couche de base de données qui doit rarement être mise à jour, vous pouvez définir une politique de pile qui empêche la plupart des utilisateurs de mettre à jour cette couche. Pour plus d'informations, consultez Empêchement des mises à jour des ressources de la pile .	2010-05-15
Nom des ressources	8 novembre 2010	Au lieu d'utiliser des ressources physiques CloudFormation générées par IDs -generated, vous pouvez attribuer des noms à certaines ressources. Les CloudFormation ressources suivantes prennent en charge la dénomination • CloudWatch Alarmes Amazon • Tables DynamoDB • AWS Elastic Beanstalk applications et environnements • Compartiments Amazon S3 • Rubriques Amazon SNS • Files d'attente Amazon SQS Pour plus d'informations, consultez Type de nom.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Attribution de types de ressources personnalisées	8 novembre 2010	Dans vos modèles, vous pouvez spécifier votre propre type de ressource pour les ressources personnalisées CloudFormation (<code>AWS::CloudFormation::CustomResource</code>). En utilisant votre propre nom de type de ressource personnalisée, vous pouvez rapidement identifier les types de ressources personnalisées que vous avez dans votre pile. Par exemple, vous pouvez définir "Type": "Custom:: <i>MyCustomResource</i> ". Pour de plus amples informations, veuillez consulter AWS::CloudFormation::CustomResource .	2010-05-15
Ajout des pseudo-paramètres	8 novembre 2010	Vous pouvez désormais faire référence à l' AWS AccountID CloudFormation dans les modèles en vous référant au <code>AWS::AccountId</code> pseudo paramètre. Pour plus d'informations, consultez Référence des pseudo-paramètres .	2010-05-15
Spécification des piles dans les politiques IAM	8 novembre 2010	Vous pouvez autoriser ou refuser à des utilisateurs, des groupes ou des rôles IAM d'opérer sur des CloudFormation piles spécifiques. Par exemple, vous pouvez refuser l'action de suppression de pile au niveau d'un ID de pile spécifique. Pour plus d'informations, consultez Controlling Access with AWS Identity and Access Management .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Prise en charge de la fédération	14 octobre 2013	CloudFormation prend en charge les informations d'identification de sécurité temporaires issues des rôles IAM, qui permettent des scénarios tels que la fédération et l'authentification unique au. AWS Management Console Vous pouvez également passer des appels CloudFormation depuis des instances EC2 sans intégrer d'informations d'identification de sécurité à long terme en utilisant des rôles IAM. Pour plus d'informations sur CloudFormation l'IAM, consultez Controlling Access with AWS Identity and Access Management .	2010-05-15
Prise en charge de réplica en lecture Amazon RDS	24 septembre 2013	Vous pouvez maintenant créer des réplicas en lecture Amazon RDS à partir d'une instance de base de données source. Pour plus d'informations, consultez la propriété <code>SourceDBInstanceIdentifier</code> dans la ressource AWS::RDS::DBInstance .	2010-05-15
Association de l'adresse IP publique aux instances d'un groupe Auto Scaling	19 septembre 2013	Vous pouvez maintenant associer les adresses IP publiques aux instances d'un groupe Auto Scaling. Pour de plus amples informations, veuillez consulter AWS::AutoScaling::LaunchConfiguration .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Prise en charge de VPC supplémentaire	17 septembre 2013	CloudFormation ajoute plusieurs améliorations pour prendre en charge les fonctionnalités VPC et VPN • Vous pouvez associer une adresse IP publique et plusieurs adresses IP privées aux interfaces réseau Amazon EC2. Pour de plus amples informations, veuillez consulter AWS::EC2::NetworkInterface. Vous pouvez également associer une adresse IP privée principale à une adresse IP Elastic (EIP). • Vous pouvez activer la prise en charge DNS et définir des noms d'hôte DNS. Pour de plus amples informations, veuillez consulter AWS::EC2::VPC. • Vous pouvez spécifier un routage statique entre une passerelle réseau privé virtuel et votre passerelle VPN. Pour de plus amples informations, veuillez consulter AWS::EC2::VPNConnectionRoute.	2010-05-15
Support des groupes de sécurité Redis et VPC pour Amazon ElastiCache	3 septembre 2013	Vous pouvez désormais spécifier Redis comme moteur de cache pour un cluster Amazon ElastiCache (ElastiCache). Vous pouvez également désormais attribuer des groupes de sécurité VPC aux ElastiCache clusters. Pour de plus amples informations, veuillez consulter AWS::ElastiCache::CacheCluster.	2010-05-15
Création, mise à jour et suppression des piles en parallèle, et mise à jour des piles imbriquées	12 août 2013	AWS CloudFormation crée, met à jour et supprime désormais des ressources en parallèle, améliorant ainsi les performances des opérations. Si vous mettez à jour un modèle de niveau supérieur, AWS CloudFormation met à jour automatiquement les piles imbriquées qui ont changé. Pour plus d'informations, consultez CloudFormation Stacks Updates.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Les groupes de sécurité VPC peuvent désormais être définis dans des instances DB RDS	28 février 201	Vous pouvez maintenant assigner des groupes de sécurité VPC à une instance de base de données RDS avec CloudFormation. Pour plus d'informations, consultez la propriété VPCSecurityGroups dans AWS::RDS::DBInstance .	2010-05-15
Déploiements propagés pour les groupes Amazon EC2 Auto Scaling	20 février 201	CloudFormation prend désormais en charge les politiques de mise à jour sur les groupes Amazon EC2 Auto Scaling, qui décrivent comment les instances du groupe Amazon EC2 Auto Scaling sont remplacées ou modifiées lorsque le groupe Amazon EC2 Auto Scaling ajoute ou supprime des instances. Vous pouvez modifier ces paramètres au moment de la création ou de la mise à jour de la pile. Pour plus d'informations et un exemple, voir UpdatePolicyAttribute .	2010-05-15
Action d'annulation et de restauration des mises à jour d'une pile	20 février 201	CloudFormation permet d'annuler une mise à jour de la pile. La pile doit avoir l'état UPDATE_IN_PROGRESS lorsque la requête de mise à jour est effectuée. Des informations supplémentaires sont disponibles dans les rubriques suivantes : • Annulation d'une mise à jour de la pile • cancel-update-stack • CancelUpdateStack dans la Référence d'API CloudFormation	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Instances optimisées EBS pour les groupes Amazon EC2 Auto Scaling	20 février 201	Vous pouvez maintenant mettre en service des instances optimisées pour EBS dans les groupes Amazon EC2 Auto Scaling pour offrir un débit dédié à Amazon Elastic Block Store (Amazon EBS) dans les instances dont la mise à l'échelle est automatique. La mise en œuvre est similaire à celle de la version précédente de prise en charge des instances EC2 optimisées Amazon EBS. Pour plus d'informations, consultez la nouvelle propriété <code>EbsOptimized</code> dans AWS::AutoScaling::LaunchConfiguration.	2010-05-15
Nouvelle documentation	21 décembre	AWS::EC2::Instance offre maintenant une propriété <code>BlockDeviceMappings</code> pour vous permettre de définir des mappages de périphérique de stockage en mode bloc pour votre instance EC2. Avec cette modification, deux nouveaux types ont été ajoutés : • BlockDeviceMapping • Périphérique de stockage en mode bloc	2010-05-15
Nouvelle documentation	21 décembre	De nouvelles sections ont été ajoutées pour décrire les procédures de création et d'affichage des piles à l'aide de l'interface remaniée d'AWS Management Console. Vous les trouverez ici : • Création d'une pile depuis la CloudFormation console • Afficher les informations relatives à la pile depuis la CloudFormation console	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle documentation	15 novembre	Des informations sur les ressources personnalisées sont fournies dans les rubriques suivantes : • Personnalisation des ressources • AWS::CloudFormation::CustomResource • Référence sur les ressources personnalisées	2010-05-15
Documentation mise à jour	15 novembre	CloudFormation prend désormais en charge la spécification I/O des opérations provisionnées par seconde (IOPS) pour les instances de base de données RDS. Vous pouvez définir une valeur comprise entre 1 000 et 10 000 (par incrément de 1 000 IOPS) via la nouvelle propriété IOPS dans AWS::RDS::DBInstance . Pour plus d'informations sur la spécification d'IOPS pour les instances DB RDS, consultez IOPS provisionnés dans le Guide de l'utilisateur Amazon Relational Database Service.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Contenu nouveau et mis à jour	27 août 2012	Les rubriques ont été réorganisées afin de fournir plus clairement des informations spécifiques sur l'utilisation AWS Management Console et l'utilisation de l'interface de ligne de CloudFormation commande (CLI). Des informations sur le balisage des CloudFormation piles ont été ajoutées, notamment de nouveaux guides et des rubriques de référence mises à jour : • Nouvelle rubrique : Configuration des options de pile. • Nouvelles informations sur les balises dans le document Référence d'API CloudFormation : CreateStack, Stack et Tag. Nouvelles informations sur l'utilisation des piles Windows : • Microsoft Windows Amazon Machine Images (AMIs) et CloudFormation modèles • Amorçage de piles CloudFormation Windows Nouveau sujet : Utilisation d'expressions régulières dans les CloudFormation modèles.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle fonctionnalité	25 avril 2012	CloudFormation fournit désormais un support complet pour la sécurité du Virtual Private Cloud (VPC) avec Amazon EC2. Vous pouvez désormais créer et remplir un VPC entier avec tous les types de ressources VPC (sous-réseaux, passerelles, ACLs réseau, tables de routage, etc.) à l'aide d'un modèle unique. CloudFormation La documentation pour les types de ressources suivants a été mise à jour : • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer De nouveaux types de ressources ont été ajoutés dans la documentation : • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		• AWS::EC2::Subnet • AWS::EC2::VPNGateway • AWS::EC2::CustomerGateway	
Nouvelle fonctionnalité	13 avril 2012	CloudFormation permet désormais d'ajouter ou de supprimer des éléments d'une pile lors de sa mise à jour. CloudFormation Les mises à jour de Stacks ont été mises à jour et une nouvelle section a été ajoutée à la procédure pas à pas : Change the Stack's Resources, qui décrit comment ajouter et supprimer des ressources lors de la mise à jour de la pile.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouvelle fonctionnalité	2 février 2012	CloudFormation fournit désormais un support pour les ressources d'un Amazon Virtual Private Cloud (Amazon VPC) existant. Avec cette version, vous pouvez : • Lancer une instance EC2 dédiée dans un Amazon VPC existant. Pour de plus amples informations, veuillez consulter AWS::EC2::Instance. • Définir l'attribut <code>SourceDestCheck</code> d'une instance EC2 qui réside dans un Amazon VPC existant. Pour de plus amples informations, veuillez consulter AWS::EC2::Instance. • Créer des adresses IP Elastic dans un Amazon VPC existant. Pour de plus amples informations, veuillez consulter AWS::EC2::EIP. • CloudFormation À utiliser pour créer des groupes de sécurité Amazon VPC et des règles d'entrée/sortie dans un VPC existant. Pour de plus amples informations, veuillez consulter AWS::EC2::SecurityGroup. • Associer un groupe Auto Scaling à un Amazon VPC existant en définissant la propriété <code>VPCZoneId</code> de votre ressource <code>AWS::AutoScaling::AutoScalingGroup</code>. Pour de plus amples informations, veuillez consulter AWS::AutoScaling::AutoScalingGroup. • Attacher un équilibreur de charge Elastic Load Balancing à un sous-réseau Amazon VPC et créer des groupes de sécurité pour cet équilibreur de charge. Pour de plus amples informations, veuillez consulter AWS::ElasticLoadBalancing::LoadBalancer.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
		Créer une instance DB RDS dans un Amazon VPC existant. Pour de plus amples informations, veuillez consulter AWS::RDS::DBInstance.	
Nouvelle fonctionnalité	2 février 2012	Vous pouvez maintenant mettre à jour les propriétés des ressources suivantes dans une pile existante : AWS::EC2::SecurityGroupIngress AWS::EC2::SecurityGroupEgress AWS::EC2::EIPAssociation AWS::RDS::DBSubnetGroup AWS::RDS::DBSecurityGroup AWS::RDS::DBSecurityGroupIngress AWS::Route53::RecordSetGroup Pour une liste complète des ressources actualisables et des informations sur les éléments à prendre en compte lors de la mise à jour d'une pile, consultez la section Mises à jour CloudFormation des piles.	2010-05-15
Restructuration du manuel	2 février 2012	Réorganisation des sections existantes en nouvelles sections : utilisation des CloudFormation modèles et gestion des piles. Déplacement de Référence de modèles en haut de la table des matières. Déplacement de l'estimation du coût de votre CloudFormation pile dans la section Getting Started.	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouveau contenu	2 février 2012	Nouvelles sections ajoutées : • Procédure de mise à jour d'une pile est un didacticiel qui explique le processus de mise à jour d'une pile LAMP. • Déploiement d'applications sur Amazon EC2 AWS CloudFormation décrit comment utiliser des scripts d'CloudFormation assistance pour déployer des applications à l'aide des métadonnées stockées dans votre modèle. • CloudFormationHelper Scripts Reference fournit du matériel de référence pour les scripts d'CloudFormation assistance (cfn-init cfn-get-metadata, cfn-signal et cfn-hup).	2010-05-15
Nouvelle fonctionnalité	26 mai 2011	CloudFormation fournit désormais la <code>list-stacks</code> commande, qui vous permet de répertorier les piles filtrées par état de pile. Les piles supprimées sont consultables jusqu'à 90 jours après leur suppression. Pour plus d'informations, consultez Description et liste des piles .	2010-05-15
Nouvelles fonctionnalités	26 mai 2011	Les commandes <code>describe-stack-resources</code> et <code>get-template</code> permettent désormais d'obtenir des informations à partir de piles qui ont été supprimées au cours des 90 jours derniers. Pour plus d'informations, consultez Liste des ressources et Extraction d'un modèle .	2010-05-15

Modifier	Date de parution	Description	Version de l'API
Nouveau lien	1 mars 2011	CloudFormation les informations relatives aux points de terminaison se trouvent désormais dans le Référence s générales AWS. Pour de plus amples informations, veuillez consulter Régions et Points de terminaison dans le document AWS General Reference .	2010-05-15
Première version	25 février 201	La sortie publique initiale de CloudFormation.	2010-05-15

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.